

CONTENTS

	<u>Page</u>	
1.0 INTRODUCTION	11	1/A11
1.1 Purpose of the Manual	12	1/A12
1.2 Organization of the Manual	12	1/A12
1.3 Conventions Used in the Manual	13	1/A13
2.0 OVERVIEW	17	1/B3
2.1 Initial Overview of the System of Programs	18	1/B4
2.2 Initial Overview of the Fixed Site Neutralization Model	20	1/B6
2.3 Extended Overview of the Fixed Site Neutralization Model	28	1/B14
2.3.1 Site Representation	28	1/B14
2.3.2 Entities Represented	32	1/C4
2.3.3 Activities Represented	38	1/C10
2.3.4 Representation of Perceptual Processes	44	1/D4
2.3.5 Representation of Decision Processes	53	1/D14
2.3.6 Representation of Communications	60	1/E7
2.4 Overview of Input Requirements	62	1/E9
2.5 Overview of System Outputs	66	
2.6 Resource Requirements for the Use of the System	74	1/E13
		1/F11
3.0 PLANNING THE RUNS	75	1/F12
3.1 Planning Plex Preprocessor Runs	76	1/F13
3.2 Planning Data Preprocessor Runs	78	1/G1
3.3 Planning Fixed Site Neutralization Model Runs	79	1/G2
4.0 USING THE PLEX PREPROCESSOR	83	1/G6
4.1 Preparing Plex Preprocessor Inputs	83	1/G6
4.2 Running the Plex Preprocessor	87	1/G10
4.3 Responding to Plex Preprocessor Outputs	88	1/G11
5.0 USING THE DATA PREPROCESSOR	91	1/G14
5.1 Preparing Data Preprocessor Inputs	91	1/G14
5.2 Running the Data Preprocessor	92	2/A2
5.3 Responding to Data Preprocessor Outputs	93	2/A3
6.0 USING THE FIXED SITE NEUTRALIZATION MODEL	95	2/A5
6.1 Preparing the Fixed Site Neutralization Model Inputs	95	2/A5
6.2 Running the Fixed Site Neutralization Model	96	2/A6
6.3 Responding to Fixed Site Neutralization Model Outputs	97	2/A7

JUN 3 1980
NUREG/CR-1307

SAND79-2241

Unlimited Release

JUN 3 1980
JUN 2 1980

43.N88:25/1307

COMPLETED

ORIGINAL

Fixed Site Neutralization Model User's Manual

Dennis Engi, Leon D. Chapman, Sandia Laboratories
W. Judnick, R. Blum, L. Broegler, J. Lenz,
A. Weintraub, D. Ballard, Vector Research, Inc.

Prepared for Sandia Laboratories, Albuquerque, New Mexico 87105
under contract to the United States Department
of Energy under contract DE-AC04-79SF00029
Printed December 1979



Sandia Laboratories

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

The views expressed in this report are not necessarily those of the U. S. Nuclear Regulatory Commission

Available from
National Technical Information Service
Springfield, Virginia 22161

SAND79-2241
Unlimited Release

FIXED SITE NEUTRALIZATION MODEL
USER'S MANUAL

Dennis Engi, Leon Chapman
Sandia Laboratories
Albuquerque, NM 87185

W. Judnick, R. Plum, L. Broegler,
J. Lenz, A. Weintraub, D. Ballard
Vector Research, Incorporated
Ann Arbor, Michigan 48106

Date Published: December 1979

Submitted by
Vector Research
Ann Arbor, Michigan 48106
under
Contract Document Nos. 07-5018 and 07-7098
to
Sandia Laboratories
Albuquerque, New Mexico 87185
operated by
Sandia Corporation
for the
U. S. Department of Energy

Prepared for
Division of Safeguards, Fuel Cycle and Environmental Research
Office of Nuclear Regulatory Research
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-550-75
NRC FIN No. A1060

BLANK PAGE

ACKNOWLEDGMENTS

This manual was prepared under contract number 07-7098 for Sandia Laboratories, Albuquerque, New Mexico, during the period 12 December 1977 to 12 June 1978. The authors wish to acknowledge gratefully the assistance of Dr. Dennis Engi, the sponsor's technical monitor.

BLANK PAGE

PREFACE

This three volume report describes the Fixed Site Neutralization Model (FSNM) which was developed by Vector Research, Incorporated, under contract to Sandia Laboratories. The report is comprised of three volumes: User's Manual, Programmer's Manual Volume I, and Programmer's Manual Volume II (Listings). The User's Manual presents a general overview of FSNM and provides instructions and guidance on the use of the model. The Programmer's Manual provides a detailed description of FSNM and preprocessor program logic and data structures.

BLANK PAGE

EXHIBITS

<u>Number</u>	<u>Title</u>	<u>Page</u>
2-1	Information Flows Within the System of Programs	19
2-2	Macroflowchart of the Fixed Site Neutralization Model	22
2-3	Visualization of the Site Geometry	33
2-4	Visualization of Building Geometry	34
2-5	Sample Parameters Describing Activity Subtypes	39
2-6	Sample Values of PSIG for a Variety of Weapons	42
2-7	Sample Values for Cross-Sectional Areas	45
2-8	Sample Priorities for Objects and Their Attributes	49
2-9	Sample Values for the SENRNG and DETSEN Arrays	52
2-10	Sample Values Determining Relative Danger of Situations	58
2-11	Site and Force Complexity Provided for in the Current Dimensions	64
2-12	Major Types of Information Input to the Data Preprocessor	65
2-13	Excerpt from Report File on Logical Device Number 6	67
2-14	Excerpts from the Final File on Logical Device Number 8	69
2-15	Excerpts from the Changes File on Logical Device Number 6	71
4-1	Documentation of File RECDESCRIP	84

1.0 INTRODUCTION

The Fixed Site Neutralization Model (FSNM) is a stochastic, time-stepped simulation of an engagement process whereby an adversary force attempts to steal or sabotage sensitive (e.g., nuclear) materials being guarded by a security force on a fixed site and a response force that is off-site. It is anticipated that the FSNM will assist regulatory bodies of the US Government in evaluating fixed site physical protection systems at various installations in a variety of scenarios. In resolution, the model has representations of individual activities, plans, perceptions, psychological profiles, skills, and equipment. The forces simulated may involve as many as 30 individuals.

For purposes of efficiency, most data input to the Fixed Site Neutralization Model are unformatted--i.e., in the machine-readable form. The user's data are transformed into unformatted representations by two supporting modules, the Data Preprocessor (DPP) and the Plex Preprocessor (PPP). The DPP handles those data to be stored in the array-type data structures most frequently associated with FORTRAN code; for the most part, it deals with performance data. The PPP loads the scenario-related data--such as site and force descriptions--into a more flexible plex-type data structure, which accommodates the use of lists and pointers in the FORTRAN code.

1.1 Purpose of the Manual

The purpose of this user's manual is to provide a general description of the Fixed Site Neutralization Model and its preprocessors, and guidance for their use on the AMDAHL 470V/6 central computing facilities of the University of Michigan, Ann Arbor. The manual is directed to those persons charged with one or more of the following responsibilities: planning of runs for a study; preparation of system inputs; running the programs; or interpreting the system outputs. It also intended for those individuals who require an overview of the FSNM and its supporting modules. Detailed information for computer system personnel is located in the program maintenance manual [VRI, 1978] of this series.

1.2 Organization of the Manual

This manual is organized into six chapters, followed by a references section and an index. Chapter 1 has introductory material, including a description of the conventions used throughout the manual. All overview materials are in chapter 2, at two levels of detail: a brief initial overview in sections 2.1 and 2.2, and an extended overview constituting the remainder of the chapter.

Chapter 3 is written for the user who wishes to plan the runs needed to accomplish a study. A careful reading of this material should increase the likelihood of productive runs and promote the efficient use of resources throughout the study.

Chapters 4 through 6 are each oriented toward the use of a specific program in the system: the Plex Preprocessor, the Data Preprocessor, and the Fixed Site Neutralization Model, respectively. These chapters are organized similarly, with three major sections treating the chronological sequence of input preparation, running the program, and output interpretation. The input preparation sections give the definitions of the variables, any prescribed formats, and information concerning the data structures involved. The running the program sections display sample job control language which will cause the execution of the programs in either batch or remote mode, according to the requirements of the Michigan Terminal System (MTS). The output interpretation sections display sample outputs and error messages and appropriate discussions.

Readers interested in perusing the FORTRAN code of the programs and/or detailed verbal descriptions of the logic are referred to the program maintenance manual [VRI, 1978].

1.3 Conventions Used in the Manual

The flowcharting conventions employed in this manual are as follows: rectangular shapes for calculations; diamonds for branching points; flattened ovals for starts, stops, entries, and returns; hexagons for the beginnings of loops; stylized print pages for output; circles for off-page connectors; and circles with underruling for data files (such as discs or magnetic tape).

In the naming of variables, subroutines, common blocks, and the like, an effort has been made to use letter combinations of mnemonic value. These names tend to emphasize purpose (especially in the case of subroutines), definitions (variables), or content (plex records). For example, the following are subroutines in the Fixed Site Neutralization Model: LDRPLN (LeaDeR PLaNning); NEWPCP (NEW PerCePtion); and ARRIVL (ARRIVaL of response force). The following are typical variable names: FORCES, summarizing the status of all players on each side; TMRESP, time the response force is to arrive; and INTCPT, indicating whether or not a communication will be intercepted. Some example plex records are: BUILDING, a record type used to describe each building on the site; and YARD, a record type used to describe each area that is external to all buildings on a site. The user who anticipates close contact with programmers during his study will find these mnemonics useful, and should take the time to note them when they are first encountered.

As a rule, this manual attempts a minimal number of references to subscripts contained in the code. When describing input preparation for subscripted variables, however, such references become necessary and are accomplished in the following manner. The existence of one or more subscripts is indicated by a pair of parentheses immediately following the name--e.g., FORCES(). The subscripts are then given a descriptive, shorthand, English form which is more readily interpreted than the relatively cryptic FORTRAN version--e.g., FORCES(PHYSICAL STATUS CATEGORY, SIDE) will be used instead of the FORCES(IPHS,IIT) that occurs in the code. Commas are used within the parentheses to separate two or more subscripts.

When inputting values for variables having more than one subscript, the user should be aware of the FORTRAN convention of varying the leftmost subscripts most quickly. To illustrate: `FORCES(1,1)` is followed by `FORCES(2,1)`, then `FORCES(3,1)`, etc., until the first side (the security forces) is finished; then comes `FORCES(1,2)`, `FORCES(2,2)`, etc., until the second side is finished. The numbers for the first subscript in these examples encode the various possible PHYSICAL STATUS CATEGORIES--such as dead = 1, captured = 2, wounded = 3, and whole = 4; these interpretations are subject to user control, but should only be varied with the assistance of a programmer.

An asterisk occasionally will be found in the English language version of a subscript--e.g., `RANGE CLASS * STD DEV`. These asterisks may be interpreted as commas in all cases, and are subject to the same remarks as above.

In dealing with programmers, the user may also find it convenient to know that most constants appearing in the program are named; these are called "named constants". To continue with the `FORCES` example, the current number of possible PHYSICAL STATUS CATEGORIES is 4. Instead of using the number 4 in the logic whenever this fact is required, the named constant `NPHCAT` is used instead. This allows for efficient change should the program be modified. Suppose that a fifth category were to be introduced, for example, Then, instead of changing 4s to 5s at the appropriate places in all the programs, one could simply assign the value of 5 to `NPHCAT` in *one* input statement. While it is unlikely that a user would ever get involved in the details of effecting such changes, he nonetheless should appreciate that very few numbers are "hard-wired" into the program allowing the model to be readily adapted to a wide variety of scenarios--one of the Fixed Site Neutralization Model's more important features.

BLANK PAGE

2.0 OVERVIEW

This chapter addresses the potential users of the Fixed Site Neutralization Model and others wishing to acquaint themselves with the capabilities and requirements of that model. An initial overview of the system of programs--i.e., the FSNM and its two preprocessors -- is in section 2.1. That section acquaints the reader with the three programs constituting the system--what they do, in what sequence they are called, what files are produced, and the general content of those files. Section 2.2 provides an initial overview of the Fixed Site Neutralization Model. A macroflowchart summarizes the logic of the engagement simulation, and is accompanied by text describing what is represented in those computations. These two sections will provide the first-time reader with the foundation necessary to read any or all of the subsequent sections of this chapter.

Section 2.3 extends the overview of the FSNM. The site representation is explained in subsection 2.3.1. The entities represented--such as people and sensors--are discussed in subsection 2.3.2. The representation of activities is detailed in subsection 2.3.3. The representation of perceptual processes is treated in subsection 2.3.4. Decision processes are the subjects of subsection 2.3.5. The extended overview concludes with subsection 2.3.6 on communications.

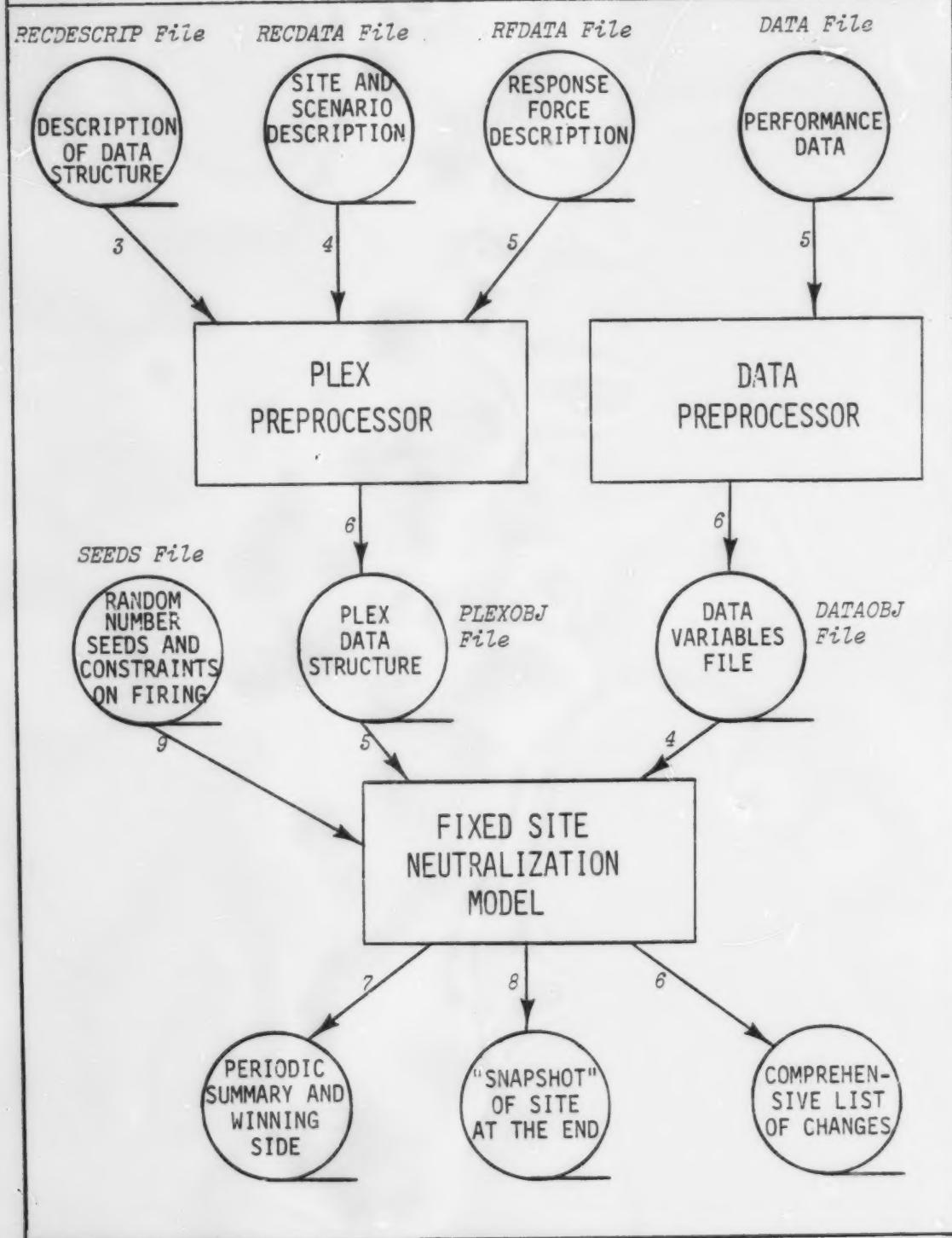
The three sections which conclude this chapter are designed to insure that the potential user has a correct understanding of the input and resource requirements necessary for the successful use of the system (sections 2.4 and 2.6, respectively), as well as an overview of the outputs which can be expected (section 2.5). Practical considerations -- such as personnel time to prepare the data base, computer time charges, and storage requirements -- are discussed

2.1 Initial Overview of the System of Programs

The purpose of this system of programs is to output, in various levels of detail, the results of a simulated engagement process whereby an adversary force attempts to steal or sabotage sensitive nuclear materials (SNM) being guarded by a security force on a fixed site and a response force that is off-site. To do this, three programs are provided: the Fixed Site Neutralization Model, which produces the engagement results; and the programs which preprocess virtually all of its inputs, the Data Preprocessor and the Plex Preprocessor.

Exhibit 2-1 shows the information flows to and from these programs and their interrelationships. (Numbers shown are logical device numbers.) In sequence, the runs of the preprocessors must come first. The files they produce are unformatted and are stored on disc for use as input files on one or more FSNM runs. The advantage of having these preprocessors is that the

EXHIBIT 2-1: INFORMATION FLOWS WITHIN THE SYSTEM OF PROGRAMS



unformatted inputs they produce are much more efficient to read when they are in this machine-readable form. (When compared to the cost of reading formatted inputs, the costs of reading unformatted inputs are negligible.) Any time one of the files prepared by a preprocessor is reused, a considerable economy results. For example, suppose that a set of FSNM runs is to study the relative value of training the guards to various sets of skill levels. Then the performance data would be changed for each FSNM run, requiring the Data Preprocessor to be run multiple times, but the site and scenario description data would only be done *once* by the Plex Preprocessor. As another example, suppose the performance data is agreed upon, and the study issue is whether a given site will be adequately defended in a variety of scenarios. Then the scenario description data would be changed for each FSNM run, requiring the Plex Preprocessor to be run multiple times, but the performance data would only be done *once* by the Data Preprocessor. The reader who has some experience dealing with computers in studies will appreciate that such avoidance of formatted input/output operations produces significant economies.

2.2 Initial Overview of the Fixed Site Neutralization Model

This section presents an overview of the Fixed Site Neutralization Model, which simulates the engagement process whereby an adversary force attempts to steal or sabotage sensitive nuclear materials being guarded

by a security force on a fixed site and a response force that is off-site. The FSNM consists of a main program, 75 subroutines, and 52 functions--all written in the FORTRAN language. Altogether, there are roughly 6,000 lines of code and 11,000 lines of comment. This initial overview will consist primarily of a macroflowchart of the logic, shown in exhibit 2-2. Many minor details of the logic are suppressed in this diagram, so as not to obscure the overall flow of logic. Only the major subroutines are referred to by name, capitalized at the top of the blocks. The reader should peruse this chart now and then read the associated commentary. Section 2.3 and its subsections may then be read for an overview that is presented in greater detail. In fact, the reader may wish to read selectively, as his interests dictate.

The FSNM is a time-stepped, stochastic simulation model. No user intervention is required during the simulation of the engagement. Because it is stochastic, random numbers are used at several points in the logic. These numbers determine courses of action selected by leaders, the detections that will occur, the firing accuracy on particular shots, and the effects of accurate shots from weapons. This gives the user the capability of assessing the overall effects (such as winning side and time to resolve the conflict) of the variability inherent in the firing, observation, and decision-making processes. That is, replications can be performed in which

EXHIBIT 2-2: MACROFLOWCHART OF THE FIXED SITE NEUTRALIZATION MODEL
(Part 1 of 3)

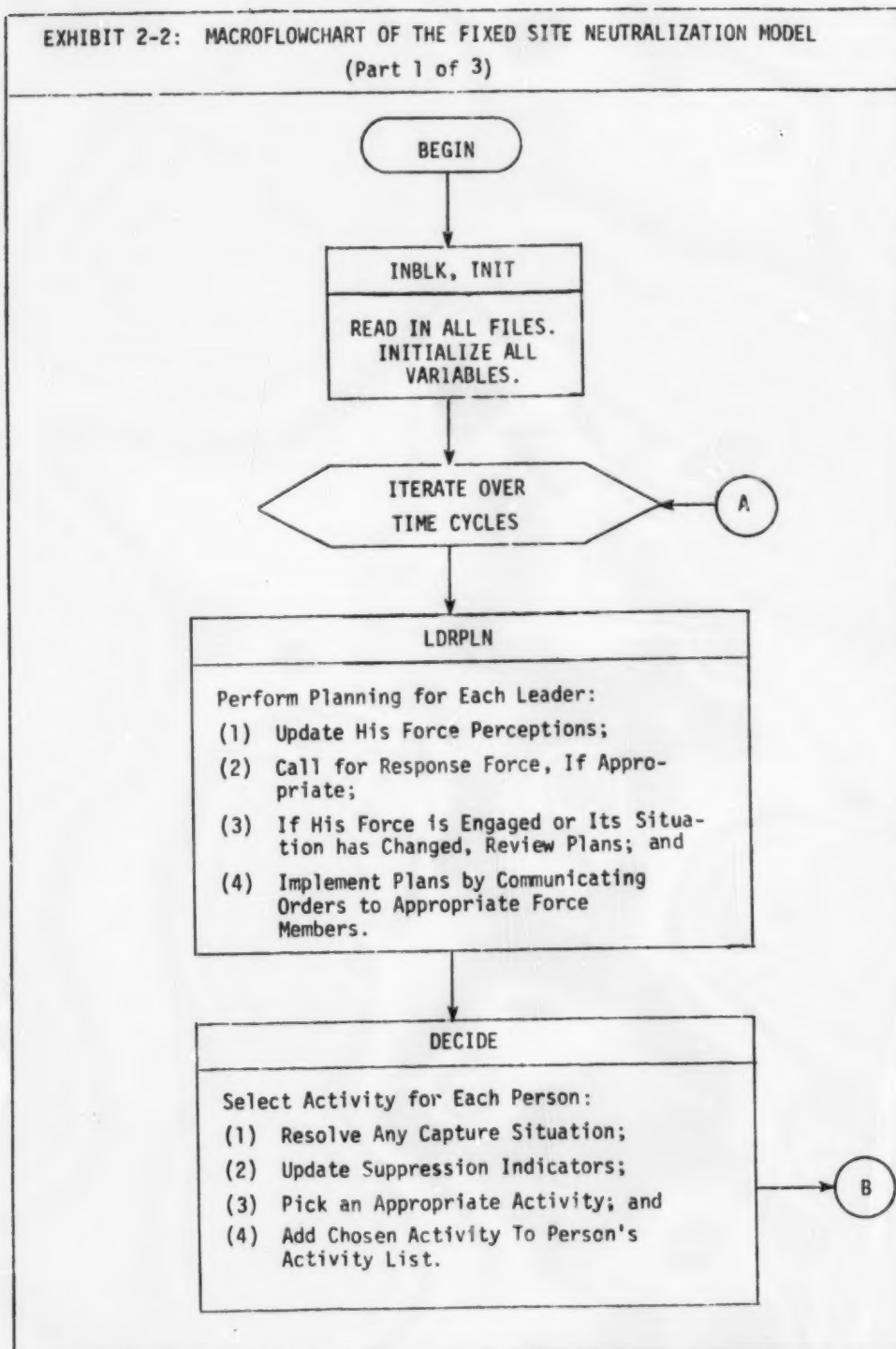


EXHIBIT 2-2: MACROFLOWCHART OF THE FIXED SITE NEUTRALIZATION MODEL
(Part 2 of 3)

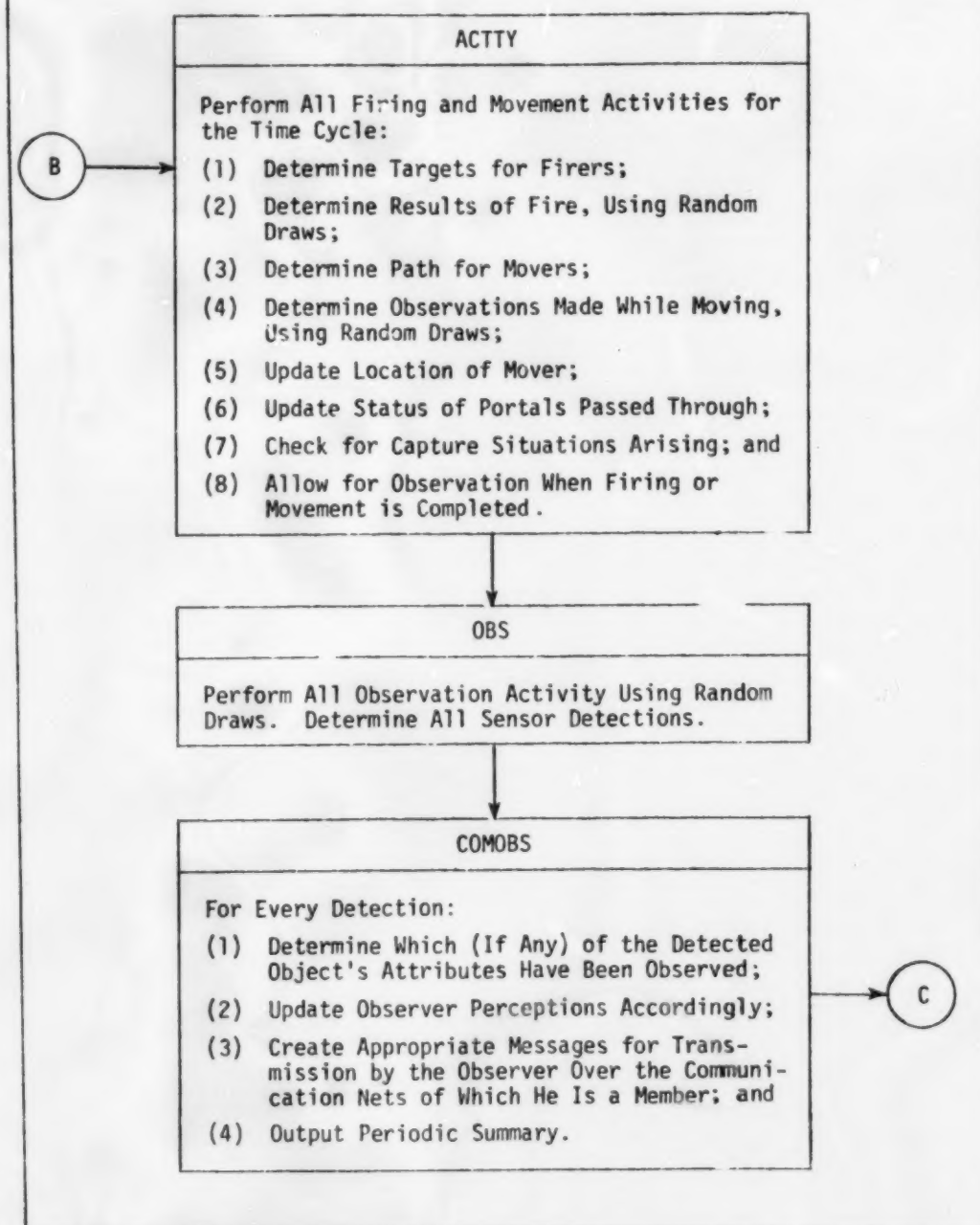
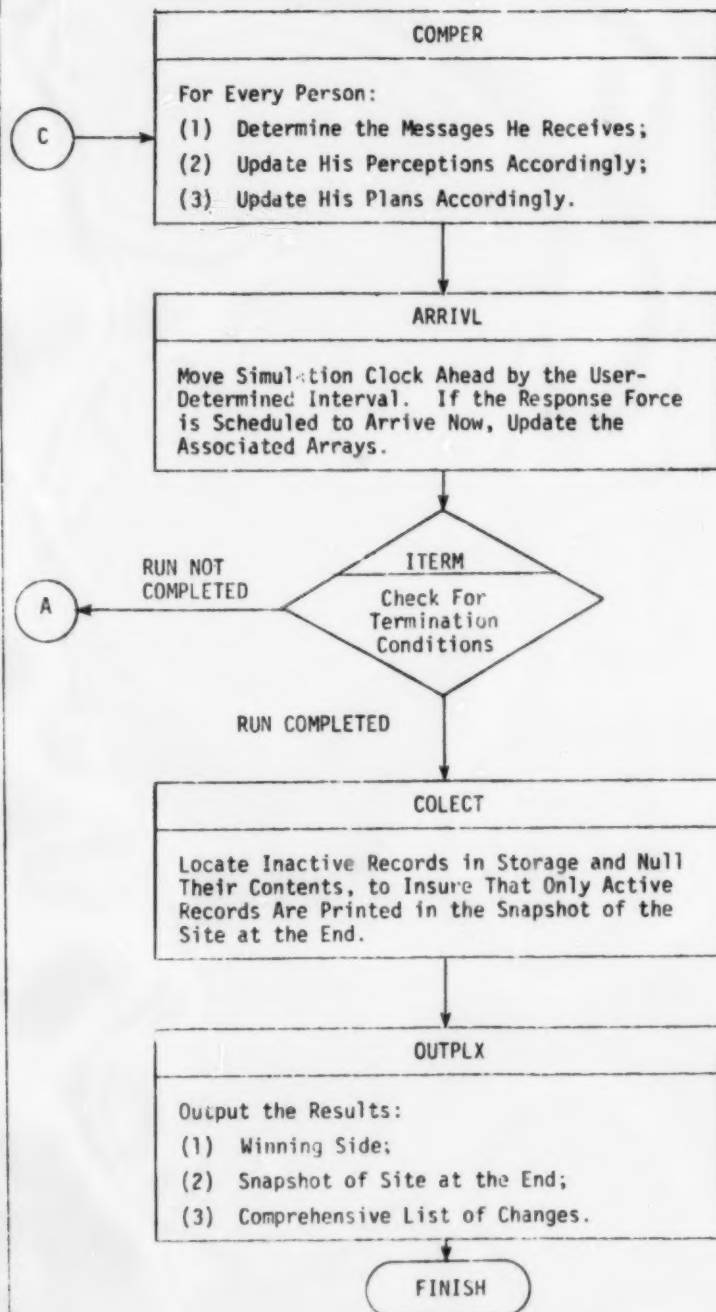


EXHIBIT 2-2: MACROFLOWCHART OF THE FIXED SITE NEUTRALIZATION MODEL
(Part 3 of 3)



each model run uses a different set of random numbers, but in which all other model inputs remain the same. Probabilities, expected values, and other statistics may be estimated from the results of such replications.

The program updates the values of all state variables at periodic intervals whose duration is under user control. Thus, it is a time-stepped simulation. Ten-second intervals have been used frequently, but the user is free to choose other intervals. Generally speaking, a short time step has the advantage of more nearly approximating the continuity of the actions and interactions taking place in the engagement; however, its concomitant disadvantage is that computing charges will be inversely proportional to the time-step duration. In practice, the user's budget may determine the appropriate balance to strike. In all cases, it is prudent to ascertain that the primary outputs of interest in a study are not unduly sensitive to the interval used. This can be established by devoting a few runs of the FSNM to shorter intervals than the one expected to be used in the bulk of the study, and then comparing their important outputs to those generated using the selected interval.

While the user-supplied time interval organizes the simulation processes in time, two geographical constructs -- "nodes" and "arcs" -- organize it in space. For every position where a person in the simulation can change his direction of motion, there is a node. Various types of nodes are distinguished, according to whether they are in

or on yards, rooms, halls, roofs, stairs, doors, or windows in the site being represented. The data describing each type of node are organized into distinctive types of records that have been named mnemonically: YARD records; ROOM records; HALL records; ROOF records; STAIRS records; DOOR records; and WINDOW records. Typically, these records provide information about the node as a point and about the region it is in as well.

Movement in the FSNM is constrained to be along straight line segments called arcs. Each arc connects two nodes; but not all pairs of nodes are connected by an arc. For example, when barriers prohibit direct passage from one node to another, there will be no such connection. As another example, two distant nodes may not be connected, if it is judged that no one would have reason to travel directly between them during the simulation. "Paths" are ordered sequences of arcs, whereby movement from an initial goal to an ultimate destination can take place by passage through intermediate nodes. The combination of representation of terrain by nodes and arcs results in a network in which every node is connected to at least one other node. These constructs allow the essential positional, dynamic, line-of-sight, and cover effects of terrain (including structures on it) to be represented economically, both in terms of storage and computation time, in the FSNM. For further details, the reader is referred to subsection 2.3.1 of the extended overview.

The physical objects or entities that are represented in the simulation are as follows: the site, its buildings, the floors of its buildings; barriers;

sensors; remote controls (activated delays); people and their weapons, equipment, and vehicles. Each of these types of entities has a distinctive record type in the plex data structure. Each entity is described by the information and contained in a record of the appropriate type. Again, mnemonic names are used for the associated records: SITE; BUILDING; FLOOR; BARRIER; SENSOR; ACTIVATED DELAY; PERSON; WEAPON; EQUIPMENT; and VEHICLE.

Intangible constructs also have corresponding record types. Every perception, possible action, message, action, relation, goal, and potential capture situation has a record of the appropriate type to describe it. The collection of all these record types, when considered as a method of organizing the data describing the site, is referred to as the "plex data structure". The employment of this type of data structure has important programming implications which will not be discussed here at any length. Basically, the plex data structure provides an important flexibility from a user's point of view. This flexibility means that the FSNM may be readily adapted to various sites and/or scenarios without being reprogrammed. When one encounters terminology associated with this structure, mnemonic names are always used and follow the same pattern: a "PERCEPT record" is the description of a perception; "MESSAGE record" is a description of a message; etc. (Particular types of information within these descriptions are called "fields", and also have mnemonic names.) This nomenclature is a convenience which should facilitate communications between the user and his programming staff.

After a careful reading of the material in this section, the reader may wish to read further in section 2.3 for an extended overview of the FSNM. The material in that section is organized into subsections, so that selective readings may be made, in accordance with the reader's interests.

2.3 Extended Overview of the Fixed Site Neutralization Model

This section presents the extended overview of the Fixed Site Neutralization Model. It assumes that the reader has read carefully the previous two sections, which present the initial overview of the system of programs and the FSNM.

This section is organized into six subsections, so that the reader may read selectively and according to his interest. Subsection 2.3.1 covers the site representation and terrain. Subsection 2.3.2 discusses the entities represented -- especially the people and how they are organized into forces. The activities represented are discussed in subsection 2.3.3. Representation of perceptual, decision, and communications processes are presented in subsections 2.3.4 through 2.3.6, respectively.

2.3.1 Site Representation

The site consists of a rectangular volume of space within which the engagement will take place. It is described by a SITE record which specifies

the following types of information: the dimensions of the site, in meters, and references to lists of buildings, barriers, yards, guards, adversaries, and vehicles on the site.

Any building on the site is assumed to be a rectangular volume. (The user will find it possible to adequately describe buildings having different shapes than this by dividing them into sets of two or more adjacent buildings, each having the prescribed shape.) Each building on the site is described in a BUILDING record, which specifies the following information: the dimensions of the building, in meters; the coordinates of the center of its bottom floor -- the basement, if it has one; and references to lists of floors, walls, and roofs which comprise its structural components.

Any barrier on the site (such as a fence or wall) is assumed to be a rectangular surface perpendicular to the ground. (Fence corners would be represented by two adjacent barriers; etc.) Each barrier on the site is described in a BARRIER record, containing the following kinds of information: the coordinates of its endpoints; its height and opacity; and references to lists of adjacent barriers and places where the barrier may be traversed (possibly following the use of explosives or wire-cutting equipment).

A yard is a node external to a building, where activities may take place or where changes of direction by moving entities may occur. The immediately surrounding region is also referred to in the term. The yard need have no particular shape, when considered as a region. Every yard on the site is

described by a YARD record, containing the following kinds of information: its coordinates; the available cover, characterized for those passing through the region and for those positioned at the node; and references to lists of its neighboring regions, the sensors which can monitor it, the barriers that are near it, and its current occupants.

The floors and roofs of the buildings on the site are described by FLOOR ROOF records, respectively. The floors and roofs are assumed to be rectangular surfaces that are parallel to the ground. Every floor is described by the following types of information: the coordinates of its center; its dimensions; a reference to the building it is contained in; and references to lists of rooms, halls, and stairs it supports. Every roof is described by the following information: the coordinates of its center; its dimensions; its penetrability; and references to lists of its neighboring regions, things on it, and sensors that can monitor it. The floor of a building may be partitioned into one or more rooms and/or halls. In either case, the assumed shape is a rectangular, planar surface. Every room and hall is described by a ROOM or a HALL record, respectively. These types of records each contain the following types of information: the coordinates of the center; the dimensions of its portion of the floor; a reference to the floor it is a part of; the amount of cover available to a person in that area; and references to lists of its contents, its neighbors, and the sensors which monitor it.

The stairways, doors, and windows on the site need not necessarily be in a building. An example of an outside stairway would be a fire escape.

Portals in barriers on the site may best be described as doors or windows. In any case, every stairs, door, and window on the site must be described by *two* STAIRS, DOOR, or WINDOW records, respectively. For stairs, the beginning and ending points must be separately described; for doors and windows, the internal and external sides are distinguished. The description of a beginning or ending of a stairway consists of the following: its coordinates; a reference to the other end; reference to the floor, yard, or roof that it is incident to; and references to lists of things located there, its neighbors, and sensors which can monitor it. The description of a side of a door has the following information: its coordinates; its dimensions; its status, lockability, penetrability, accessibility, and opacity; references to the other side and the barrier it is contained in; and references to lists of things located there, sensors that can monitor it and neighboring regions. The description of a window side is almost identical to that of door side: the only difference is that no reference to the barrier it is contained in is necessary.

The possible paths of movement on the site can be constructed from the neighboring-regions type of list referred to in the records that have just been described. To summarize, yards, rooms, halls, stairways, doors, and windows have such information in their records; therefore, the straight lines linking these neighbors are the arcs from which the paths are made. (The concept of arc was first introduced in section 2.2.)

The combination of all the information detailed above produces a site description. To illustrate how one could visualize this description, two figures are provided in exhibits 2-3 and 2-4. Exhibit 2-3 shows a portion of a hypothetical site having two buildings on it and surrounded by a fence. The various nodes are indicated with dots; their identification is indicated with a code consisting of letters and/or numbers. The yards are indicated by numbers without any letters. The doors and windows occur in pairs and are encoded DR and WI, respectively. The dashed lines connecting the nodes are the arcs indicating possible paths of movement. The physical features on the site are the buildings (BD) and barriers (BR). Notice the treatment of the guardhouse (BD2), at the normal point of entry to the site, and the two places where adversaries may force their way in through the barriers. Detail within the buildings is omitted. Exhibit 2-4 then shows a portion of building 1. Notice that the paths passing through a room (RM) go through the node at its center. Also note the treatment of a thick wall with two sets of doors dividing the building.

2.3.2 Entities Represented

There are three types of observable entities which may be present on the site: personnel, possibly organized into forces; sensors, possibly monitored and/or activated remotely; and vehicles. This subsection describes their representation in the Fixed Site Neutralization Model.

EXHIBIT 2-3: VISUALIZATION OF THE SITE GEOMETRY

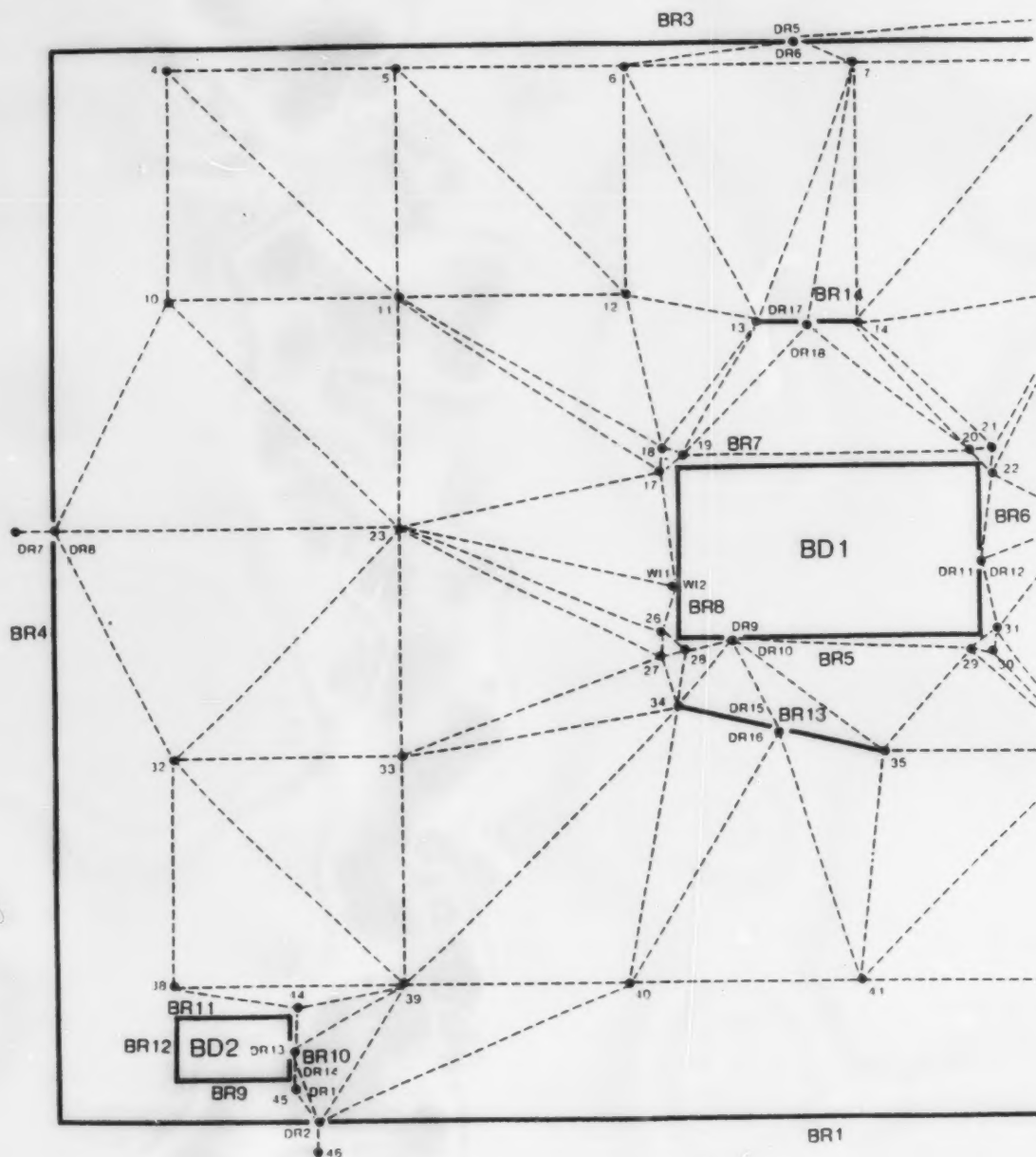
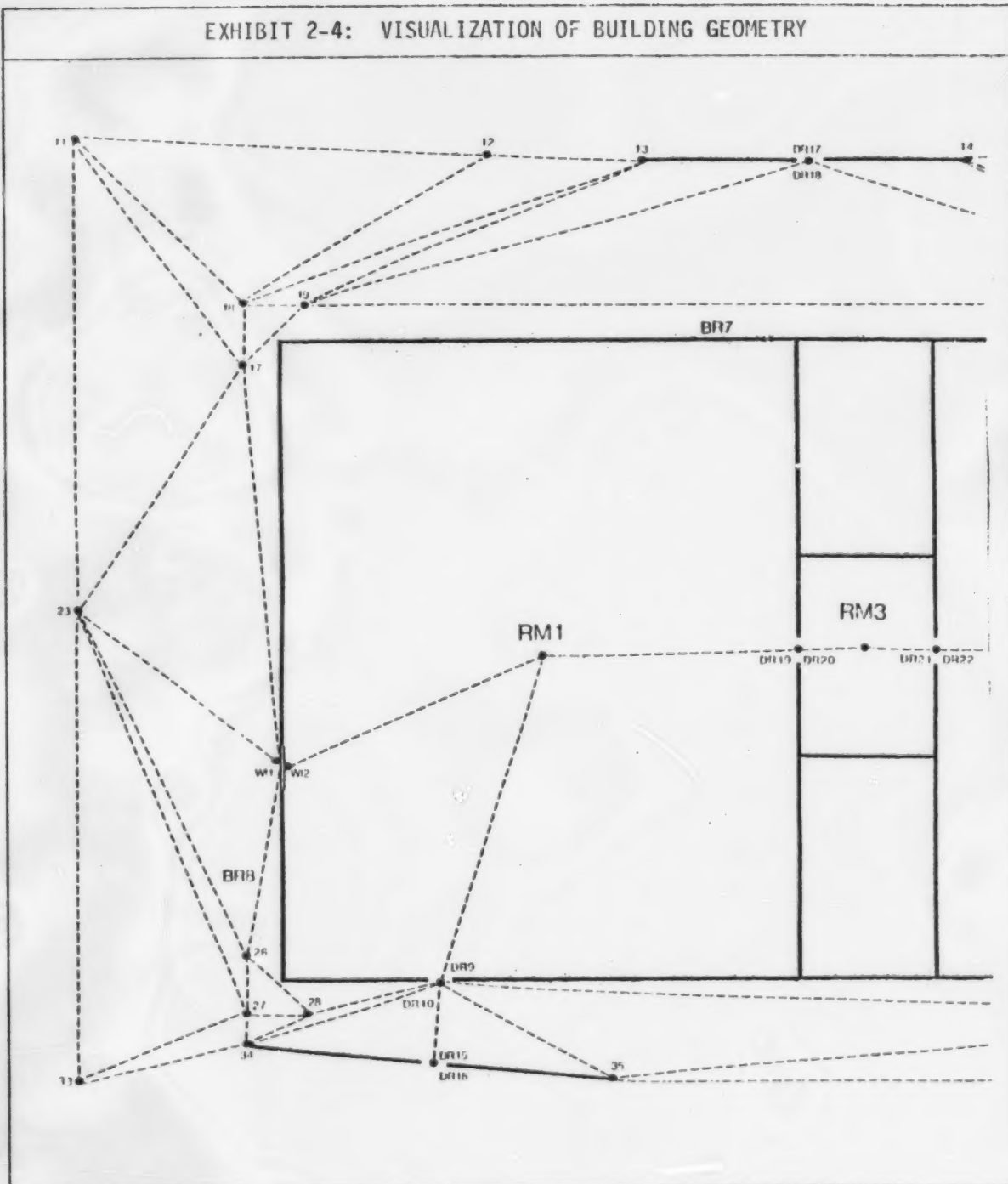


EXHIBIT 2-4: VISUALIZATION OF BUILDING GEOMETRY



Every person who is on the site initially (guards, insiders, and adversaries) or may arrive at a later time (response force members) is described by a PERSON record. The following types of information constitute a complete description of a person: type (worker, sensor monitor, combatant, or insider); physical status; allegiance; credentials; current situation; suppression code;¹ posture; time remaining for the performance of activities in the current cycle;¹ references to his location, leader, current activity,¹ and force; and references to lists of weapons and equipment he carries, his subordinates (if any), his perceptions, his planned activities,² and his standard operating procedures.² Portions of this description are optional and may be omitted on input because the program can initialize and maintain appropriate values. For example, the reference to the list of planned activities need not be input since the leader-planning logic will supply it.

The weapons and equipment carried by people are described by WEAPON and EQUIPMENT records, respectively. The description of a weapon contains the following: the type (pistol, rifle, machine gun, light antitank weapon, grenade, etc.); and the number of rounds or bursts that can be fired, considering ammunition on hand. Items of equipment are described as follows: the type (key, scaling equipment, special nuclear materials, bulletproof vest, etc.); and a descriptive parameter (access code for keys, height scalable for scaling equipment), if necessary.

¹Activities and suppression are discussed in subsection 2.3.3.

²Planning and standard operating procedures are discussed in subsection 2.3.5.

Persons may be organized into forces and/or groups, which are described in FORCE and GROUP records. Every person, except for insiders whose true allegiance is not yet known, should be a member of a force -- even if it is a one-man force -- for purposes of the leader planning logic. The force description is as follows: the allegiance (defenders, attackers); the status, or fraction of effective strength; the situation; a reference to its leader; and a reference to a list of its members, with vehicles included when appropriate. Currently, groups are not created or referred to in the demonstration version of the planning logic. However, any group of entities having a common attribute -- such as people having the same plan -- could be created and referred by other logic. If so, the GROUP record would contain the following information: the type; the number of entities in it; a references to the common attribute; and a reference to a list of group members.

Every sensor on the site is described by a SENSOR record. This description contains the following items of information: the type (explosive, metal, radioactivity, portal status, person/vehicle type 1, person/vehicle type 2, or credentials sensor); the physical status (destroyed, tripped, operational but not tripped); the type of report it generates (recording or not, binary or pictorial); references to its location and the room its monitor is in; and a list of regions it covers when operational. A sensor may be associated with a remote control that activates it. Each such remote control is described

by an ACTIVATED DELAY record containing the following information: type; physical status; a reference to its location; and a reference to its associated sensor. (While current logic provides for the processing of such information, the demonstration version of the FSNM planning routines does not include references to this type of record.)

Both sensors and persons generate messages as a result of their observations. The representation of the perceptual processes generating these observations¹ are discussed in subsection 2.3.4. The communication of messages is discussed in subsection 2.3.6.

Vehicles are not currently used in the simulation. Ultimately, the VEHICLE records will provide the capability of playing various types of vehicles and their interactions with other entities in the simulation. The Plex Preprocessor can currently input, process, and output VEHICLE records containing the following types of information: the type; physical status (destroyed, immobilized, operational); the force association; the current velocity; references to its location and driver (if any); and reference to a list of its contents, such as persons, weapons, and equipment. Moreover, provision has been made in the logic of the FSNM observation, sensor, and communication routines to deal with these records. But the demonstration version of the planning routines cannot process these descriptions, nor can the movement routines currently discriminate between persons and vehicles.

¹Individual's perceptions are fashioned from some of these observations.

2.3.3 Activities Represented

The persons in the simulation may engage in four major types of activities: moving, firing, observing, or participating in a capture situation. Within all but the latter activity are various activity subtypes, as shown in exhibit 2-5. Each activity subtype and the capturing activity itself are described by five arrays: ACTRAT, IACTTP, ISECAC, NDTSEC, and POSTUR. ACTRAT gives either a normal rate of performance, or a duration, depending upon whether the value from IACTTP is 1 (indicating meters/second as the units) or 2 (indicating seconds as the units), respectively. ISECAC flags whether the activities may be performed only in an overt fashion (code: 1) or also covertly (code: 2) by an insider. NDTSEC describes the number of octants a player can observe in, out of a maximum possible of eight; for further details, see a description of perceptual processes in subsection 2.3.4. The POSTUR array gives the proportion of the person's erect height that he will assume when performing the acts; these proportions are subject to modification when the logic considers the available cover, and should be supplied by the user on the assumption that there is no cover. Exhibit 2-5 displays sample values for these arrays for all activity subtypes and for involvement in captures.

Firing activities reference the following types of information: the description of the firing weapon, located in a WEAPON record (details are in subsection 2.3.2); the array RNGMAX, providing the maximum range in meters;

EXHIBIT 2-5: SAMPLE PARAMETERS DESCRIBING ACTIVITY SUBTYPES (Part 1 of 2)

<u>Activity Subtype</u>	<u>Activity</u>	<u>Description</u>	ACTRA1	IACOTP	ISECAC	NDTSEC	POSTUR
	1	Moving					
1		Crawling	0.45	1	2	5	0.5
2		Walking	1.25	1	2	5	1.0
3		Running	3.50	1	2	5	1.0
4		Scaling Barrier	0.45	1	2	5	0.5
5		Vehicle, normal	9.00	1	2	5	1.0
6		Vehicle, fast	18.00	1	1	5	1.0
7		Penetrate open portal	0.05	2	2	5	1.0
8		Penetrate unlocked portal	1.00	2	2	5	1.0
9		Penetrate portal with key	3.00	2	2	5	1.0
10		Penetrate portal with wpn	3.00	2	1	5	1.0

EXHIBIT 2-5: SAMPLE PARAMETERS DESCRIBING ACTIVITY SUBTYPES (Part 2 of 2)

<u>Activity Subtype</u>	<u>Activity</u>	<u>Description</u>	<u>ACTRAT</u>	<u>IACCTP</u>	<u>ISECAC</u>	<u>NDTSEC</u>	<u>POSTUR</u>
	2	Firing					
11		Pistol or rifle	3.00	2	1	3	0.2
12		Machine gun	3.00	2	1	3	0.3
13		LAW	5.00	2	1	3	0.6
14		Grenade	5.00	2	1	3	0.2
	3	Observing					
15		Stationary object	1.5	2	2	8	0.2
16		Moving object	3.0	2	2	8	0.2
17		Firer	3.0	2	2	8	0.2
18	5	Resting	30.0	2	2	0	1.0
19	4	Capturing	60.0	2	1	5	1.0

the array RNGMIN, giving the minimum range from safety considerations;
the array BASLOD, giving the basic ammunition load in appropriate units;
the array RNGLET, the lethal radius of area-fire weapons (such as grenades);
the array IFIRTP, giving the type of weapon (1 = impact lethality, 2 =
area effects); the array PK, giving the probability of kill or wound given
a hit; the array PWOUND, giving the probability of a wound, given a hit; and
the array PSIG, supplying horizontal and vertical standard deviations of the
impacts relative to the aim point, in meters. Sample PSIG values are shown
in exhibit 2-6 for a variety of weapons. Note that each type of weapon shown
has four range bands associated with it.

A person's activity or potential activity is described by an ACTION
record, containing the following types of information: the activity type
(see exhibit 2-5); three parameters, which depend on the activity type for
their interpretation; a reference (optional) to a logical condition or re-
lationship that should be satisfied on termination; and a reference to the
person performing or planning to perform the activity. The parameters for
observation are as follows: the direction of observation, in radians; a
reference to a region or list of regions to be observed; and a duration of
observation. For movement activity, there are but two parameters: a
reference to the destination and a reference to the path selected, usually
an ordered list of intermediate nodes and the final node. For firing, the

EXHIBIT 2-6: SAMPLE VALUES OF PSIG FOR A VARIETY OF WEAPONS

Weapon Type	Range Band	Standard Deviations for	
		Horizontal	Vertical/Range
1 (Pistol)	0 - 15	0.125	0.125
	15 - 30	0.20	0.20
	30 - 45	0.38	0.45
	45 - maximum	0.6	1.0
2 (Rifle)	5 - 55	0.02	0.02
	55 - 105	0.05	0.75
	105 - 155	0.10	0.15
	155 - maximum	0.30	0.45
3 (Machine Gun)	25 - 100	0.02	0.02
	100 - 175	0.15	0.20
	175 - 250	0.30	0.45
	250 - maximum	1.0	1.5
4 (LAW)	25 - 40	1.0	1.5
	40 - 55	1.2	1.8
	55 - 70	1.5	2.0
	70 - maximum	1.9	2.6
5 (Grenade)	10 - 20	1.0	1.0
	20 - 30	2.0	2.0
	30 - 40	4.0	5.0
	40 - maximum	7.0	9.0

Note: All units are in meters.

parameter interpretations are as follows: a reference to the target; a reference to the target location; and a reference to the weapon.

Human factors limitations and the variations in abilities amongst people are considered in the performance of these activities. The array SKILL -- indexed by activity type, person, and physical condition -- gives nonnegative, multiplicative correction factors that are applied to the normal rates or durations given in ACTRAT. For example, if person 1 can run ten percent faster than the norm when he is not wounded, then the factor 1.10 would be entered in the appropriate position of SKILL. Note that if ACTRAT supplies a duration, instead of a rate, then good performance would be reflected by a factor that is *less* than 1.0.

Suppression is modeled. For each person in the simulation, the user inputs four numbers, between 0. and 1. inclusive, used to define his firing threshold, his movement threshold, his observation threshold, and his surrendering threshold. (The corresponding array names are: RESFIR, RESMOV, RESOBS, and RESSUR.) A psychological indicator is kept by the model, called his "suppression level". To determine if the person is suppressed in firing, suppressed in movement, suppressed in observation, or suppressed in surrendering a simple rule is used. For the person to be able to perform the indicated activity, his suppression level must be less than one minus the activity threshold; otherwise, he is not permitted to perform the activity.

The model updates the person's suppression level using the following inputs; SUPDEC, a number subtracted from the suppression level at the

beginning of each time cycle, to permit recovery from suppression; SUPFIR, SUPMOV, and SUPOBS, which are values added to the suppression level when the logic determines that he is suppressed in firing, moving, or observation, respectively; RESPLN, which lowers the suppression level to accommodate the plan supplied to him by his leader; and RESWND, an adjustment applied only if the person is wounded. The array RISK defines a special threshold for each side, used to determine the degree of danger each is willing to risk in order to guard or to gain access to a vital area on the site -- e.g., one that has the SNM.

The decision processes used to arrive at the current and planned activities for all players are discussed in subsection 2.3.5.

2.3.4 Representation of Perceptual Processes

This subsection reviews the treatment of perceptual processes in the FSNM. These processes produce observations by individuals and detections by sensors, which are then used to produce the perceptions.

The following parameters are used to determine visual detections:

- AREA, an array specifying the average cross-sectional area of a given physical object type (for sample values, see exhibit 2-7).
- DETANG, an array which specifies the octant scan probabilities of an observer with an unobstructed full-circle field of view, beginning with his principal direction of observation (default values: .247, .211, .125, .039, .003, .039, .125, and .222).

EXHIBIT 2-7: SAMPLE VALUES FOR CROSS-SECTIONAL AREAS (Part 1 of 2)

Physical Object Type

Movable Physical Object Type

<u>Weapon Type</u>			<u>Description</u>	<u>Area (M²)</u>
1	1	1	Pistol	0.015
2	2	2	Rifle	0.1
3	3	3	Machine Gun	0.3
4	4	4	LAW	0.08
5	5	5	Grenade	0.01
6	6	6	Tear Gas	0.01
7	7	7	Nerve Gas	0.01

Equipment Type

8	8	1	Key	0.001
9	9	2	Scaling Equipment	0.1
10	10	3	Special Nuclear Material (SNM)	0.5

Shield Type

11	11	4	1	Bullet Proof Vest	0.25
12	12	5	2	Gas Mask	0.06

EXHIBIT 2-7: SAMPLE VALUES FOR CROSS-SECTIONAL AREAS (Part 2 of 2)

Physical Object Type

Movable Physical Object Type

	<u>Vehicle Type</u>			<u>Description</u>	<u>Area (M²)</u>
13	13	1	1	Car	7.0
14	14	2	2	Truck	10.0
15	15	3	3	APC	12.0
16	16	4	4	Helicopter	15.0

Person Type

17	17	5	1	Person	1.0
----	----	---	---	------------------	-----

Sensor Type

18	1	1	Explosive Sensor	0.05
19	2	2	Metal Sensor	0.05
20	3	3	Radioactivity Sensor	0.05
21	4	4	Portal Status Sensor	0.05
22	5	5	Person/Vehicle Sensor 1	0.05
23	6	6	Person/Vehicle Sensor 2	0.05
24	7	7	Credentials Sensor	0.05

Activated Delay Type

25	8	1	Door Locker	0.05
----	---	---	-----------------------	------

- DETPHI, the average vertical angle of search by an observer, in radians (default value: .321/5).
- DETRAT, an array of multipliers applied to the base detection rate to reflect the activity of the detected object.
- NDTSEC, an array specifying the number of octants the observer will scan, depending upon his current activity (defaults: one while firing, five while moving, and eight otherwise).

Also used in the representation of visual detection are the entry in the array ACTRAT which controls the amount of time in seconds necessary to recognize a target -- sometimes called the "fixation time" of the observer -- and the entries in the SKILL array which describe the relative visual acuities of the various observers.

The basic mathematical model of detection used in the model is founded upon the work of Lasken¹ and Bishop and Clark.² The probability of observer i detecting object j in an interval of duration T is:

$$p_{ij} = 1 - \exp [- 432 (\text{AREA}/\text{area searched}) p_k \text{ DETRAT } f_j T],$$

where AREA, DETRAT, and T are quantities defined above;

area searched corresponds to the number of octants specified by NDTSEC;

p_k is the probability of searching the octant k in which j lies, relative to i , a normalized probability from DETANG that takes into account the octants not searched; and

$f_j = 1$ if the object to be detected is stationary; 2, otherwise.

¹Robert A. Lasken, *Effects of Visual Acuity on Target Acquisition*, Proceedings, Military Operations Research Society, Summer 1976, pp. 112-122.

²Albert B. Bishop and Gordon M. Clark, *The Tank Weapon System*, Report AR 69-2A (U) to the US Army Combat Development Command Armor Agency, The Ohio State University, October 1969, chapter 4.

After computing this probability, the model makes a random draw from a uniform probability distribution on the $[0,1]$ interval, and compares the values. If the random number is less than or equal to p_{ij} , then i sees or detects j .

Once an object has been detected by an observer, various features or attributes of the object may or may not be recognized by him. These recognitions are controlled by parameters giving priorities for recognizing attributes. (Since detection of an object is enabled in the model by specifying the identity of the object, identity is not considered an attribute for purposes of the present discussion.) There are three types of objects that can be observed in the model -- persons, vehicles, and sensors. The priorities for observing these attributes of these types are given, respectively, in the arrays POBPSN, POBVEH, and POBSEN. POBPSN is indexed on the observer's side, the allegiance of the detected object, and the activity of the object. POBVEH depends upon the observer's side and the side owning the vehicle. POBSEN depends on the side of the observer only.

These recognition priorities are then combined with entries from the array POBATT which gives the attribute recognition priorities, to arrive at a priority for each attribute of each object detected. All priorities are given as real numbers between 0 and 1, with a high priority indicated as a number close to 1. Sample values for these arrays are displayed in exhibit 2-8.

Once the priorities of each attribute of each detected object are computed, the observer is allowed to recognize the attributes of highest

EXHIBIT 2-8: SAMPLE PRIORITIES FOR OBJECTS AND THEIR ATTRIBUTES (Part 1 of 2)

1. POBPSN (Person)

Observer = DEFENDER ATTACKER DEFENDER ATTACKER

Observed = DEFENDER DEFENDER ATTACKER ATTACKER

Activity

Moving	0.6	0.8	0.8	0.6
Firing	0.75	1.0	1.0	0.75
Observing	0.6	0.7	0.7	0.6
Capturing	0.6	0.9	0.9	0.6
Resting	0.9	0.7	0.7	0.9

2. POBSEN (Sensor)

<u>DEFENDER</u>	<u>ATTACKER</u>
0.2	0.6

3. POBVEH (Vehicle)

Side Observing

Side Owning

	<u>DEFENDER</u>	<u>ATTACKER</u>
DEFENDER	0.5	0.8
ATTACKER	0.8	0.5

**EXHIBIT 2-8: SAMPLE PRIORITIES FOR OBJECTS AND THEIR
ATTRIBUTES (Part 2 of 2)**

4. POBATT (Feature)

				<u>POBATT</u>
1	Vehicle	1	Type	0.8
2		2	Status	0.5
3		3	Location	0.9
4		4	Contents	0.0
5		5	Owner	1.0
6		6	Speed	0.3
7		7	Driver	0.0
8	Person	1	Type	0.2
9		2	Status	0.5
10		3	Location	0.9
11		4	Allegiance	0.9
12		5	Credentials	0.0
13		6	Situation	0.0
14		7	Suppression	0.5
15		8	Posture	0.5
16		9	Weapons	0.9
17		10	Leader	0.0
18		11	Subordinates	0.0
19		12	Commo Nets	0.0
20		13	Perceptions	0.0
21		14	Activity Time	0.0
22		15	Activity	0.95
23	Sensor	16	Plans	0.0
24		17	SOPs	0.0
25		18	Force	0.0
26		1	Type	0.8
27		2	Status	0.9
28		3	Location	0.95
29		4	Coverage	0.85
30		5	Reports To	0.0
31		6	Report Type	0.0

priority to him; the total number of such attributes is limited by a value from the array HFLIMS (Human Factors LIMItationS).

The observations of others may be perceived by persons after they have received messages from them. The processing of messages received is discussed in subsection 2.3.6.

The performance of remote sensors is controlled by two arrays: SENRNG, which gives the absolute maximum ranges for various types of sensors, in meters; and DETSEN, which modifies that range, depending on the movable physical object type detected. Sample values for these arrays are given in exhibit 2-9.

When a person or vehicle moves and enters the coverage area of an operational sensor which is sensitive to the person, vehicle, or the equipment carried by the person or vehicle, a message is sent to a location on the site normally occupied by a specially designated security guard known as the sensor monitor. This process is similar to the communication process described in subsection 2.3.6: the sensor monitor is considered to be on a special communications net having as its other members the sensors reporting to the sensor monitor. Once a sensor signal has been sent on this net, the status of the reporting sensor is changed to indicate that it has been tripped if the sensor is of the binary (alarm on or off) type.

After all movement has been completed, a list of new sensor activation messages is available to the sensor monitor, who processes these messages

EXHIBIT 2-9: SAMPLE VALUES FOR THE SENRNG AND DETSEN ARRAYS

Sensor Type	SENRNG	DETSN Values by Moveable Physical Object Type																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Explosives	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Metal	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.0	0.0	0.5	0.5	0.5	0.0
Radioactivity	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Portal Status	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0
Person/Vehicle	50.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.1	1.0	1.0	1.0	1.0	1.0
Person/Vehcile	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
Credentials	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Key to movable physical object types: 1 = pistol; 2 = rifle; 3 = machine gun; 4 = light antitank weapon; 5 = grenade; 6 = tear gas; 7 = nerve gas; 8 = key; 9 = scaling equipment; 10 = SNM; 11 = bulletproof vest; 12 = gas mask; 13 = automobile; 14 = truck; 15 = armored personnel carrier; 16 = helicopter; and 17 = person.

subject to limitations on the amount of sensor information he can absorb within the duration of the model's time step. Within this limit, the sensor activations are treated as direct observations by the sensor monitor, subject to the limitations of the type of sensor reporting. In addition, at this time those messages identified as coming from recording sensors are entered in a queue, together with their times of entry. A model parameter determines how long a recorded message is held in the queue before it is purged. If the sensor monitor is not saturated with new sensor activation messages, he may select older recorded messages from the queue to add to his current observations, according to a predetermined priority scheme which gives preference to the detection of adversaries.

2.3.5 Representation of Decision Processes

This subsection discusses the leader planning process and the process whereby an individual selects an activity. They are discussed in that order because this is the order in which the model simulates them at the beginning of each time interval (see parts 1 and 2 of exhibit 2-2).

The purpose of the planning process is to represent the command and control function which a leader applies to force under his direction to achieve the objectives of the force. The objectives of a leader of the adversary force are derived from the GOAL NODE records describing a sequence of nodes which his force is to occupy. The following information is provided

in one such record: time of control required at that node to perform the adversary mission successfully; time of control achieved so far at that point; minimum number of adversaries necessary at that point to achieve success; a reference to the identification of the node; a reference to a successor GOAL NODE record, if any, corresponding to the next objective; and a reference to a list of weapons and/or equipment that must be present at the node for adversary success. A security force leader's objectives are defined relative to the adversaries' objectives; he is to thwart them. The process for both leaders consists principally of the selection of plans and orders appropriate to the situation, as they perceive it.

Planning is represented in the model as a stochastic process. That is, the perception of some given tactical situation which could reasonably be expected to produce some explicit action on the part of a force leader does not necessarily result in the specification of the most desirable action. Any one of a number of plausible actions (or groups of associated actions) might be selected. The selections could range from the timid "do nothing" response, through successively more aggressive actions, to some act of reckless abandon.

A given leader or decision maker selects his action in response to a given situation through the influence of his perceptions of the situation, both as a individual and a leader, and the more subtle influence of human factors which are affecting the leader at the time of decision. Some of the

factors which a behaviorist might consider in describing the impact on a decision are: style of leadership, aggressiveness, skill in decision making, stress, and training.

The impact of human factors on a leader's decisions is represented in the model through a probability distribution governing the frequency with which the leader will select a particular action or group of associated actions in similar situations. For example, consider the following abstraction which summarizes the planning process of the model. If the leader perceives the tactical situation to be as represented by the symbol S_1 , then he has decision alternatives D_1 through D_n , each of which is defined by an explicit action or group of actions. The response of the leader to S_1 is controlled by the probability distribution which describes the relative likelihoods with which he will choose the various alternatives. In the model, he will choose one of these alternatives based on a random draw from the uniform probability distribution on the interval $[0,1]$. Each of his possible responses is uniquely associated with a segment of that interval, and the length of this segment is equal to its probability. Thus, the random draw is a number coming from exactly one of these segments, and the action associated with that segment will be chosen.

Not only does this technique capture the randomness inherent in human decision making, but also it permits the unique character of specific decision makers to be represented in the model. A further convenience derives

from the facility with which transfer of leadership can be effected -- e.g., after a leader becomes a casualty.

The leader planning subroutine (LDRPLN) of the FSNM makes extensive use of decision rules to describe the various alternatives permitted to leaders perceiving a given tactical situation. These decision rules can be defined by the user for application in a given scenario, or the user may adopt the demonstration decision rules that are already provided. The logical requirements for any new decision rules are simply stated:

- (1) maintain the global perceptions of each leader with respect to his own force and other forces in the scenario;
- (2) identify tactical situations in which responses by leaders are expected;
- (3) associate a set of decision alternatives for each tactical situation thought to be of interest; and
- (4) select a decision alternative in the stochastic manner described above.

After the leader has determined his response to the situation, appropriate messages are created to transmit his wishes to his subordinates. The transmission and reception of messages is discussed in detail in subsection 2.3.6. Suffice it to say that delays may occur before the subordinate is aware of his leader's orders, and that receipt of such messages is represented in the model by placing ACTION records at the top of a list of intended

activities associated with that subordinate. (The contents of ACTION records are detailed in subsection 2.3.3.)

The process by which individual persons select their own activities is called activity selection. Activity selection differs from leader planning in the important respect that it is a deterministic process. No random draws are made or interpreted.

Activity selection is controlled by a set of *decision rules*. For example, there is a decision rule which computes the desirability of a person firing at a potential target, given the situation in the simulation and the predilection of the person to fire. Generally speaking, the logic currently employs demonstration decision rules that are embodied in the logic of specific subroutines. The user may supply his own logic, if he wishes, taking advantage of the model's modular structure. That is, such changes need not involve extensive rewriting of model code, because the subroutines involved can be considered independently.

The decision rules and their major purposes are outlined below, in alphabetical order:

EFFTIM computes the time involved in a given person's traversal of a particular arc on his path. It uses the arrays DNGWEP and DNGDET, which provide the individual's perceived dangers of detection and exposure to fire, respectively, under various sets of circumstances. Sample values for these arrays are shown in exhibit 2-10. The flag ISECPL, which indicates the secrecy plan for side (1 = over, 2 = covert) is also consulted.

EXHIBIT 2-10: SAMPLE VALUES DETERMINING RELATIVE DANGER OF SITUATIONS

DNGDET

<u>Secrecy Plan</u>	<u>Detector Type</u>								
	<u>Manned Vehicle</u>	<u>Person</u>	<u>Sen1</u>	<u>Sen2</u>	<u>Sen3</u>	<u>Sen4</u>	<u>Sen5</u>	<u>Sen6</u>	<u>Sen7</u>
Overt (=1)	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Covert (=2)	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Key to Sensor Abbreviations: Sen1 = Explosives Sensor; Sen2 = Metal Sensor; Sen3 = Radioactivity Sensor; Sen4 = Portal Status Sensor; Sen5 = Person/Vehicle Sensor, Type 2; and Sen7 = Credentials Sensor.

DNGWEP

<u>Vehicle or Person Type</u>	<u>Weapon Type</u>							
	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>None</u>
Car	0.3	0.7	0.8	0.3	0.3	-	-	0.2
Truck	0.3	0.7	0.8	0.3	0.3	-	-	0.2
APC	0.3	0.7	0.8	0.3	0.3	-	-	0.2
Helicopter	0.3	0.7	0.8	0.3	0.3	-	-	0.2
Person	0.5	0.7	0.7	0.2	0.6	-	-	0.1

Key to Weapon Type Numberal: 1 = Pistol; 2 = Rifle; 3 = Machine Gun; 4 = Light Antitank Weapon; 5 = Grenade; 6 = Tear Gas; and 7 = Nerve Gas.

FDESIR computes the desirability of firing at a potential target. It uses the array EVALN. If the person ultimately fires, then he will fire at the target having maximum desirability.

MDESIR computes the desirability of movement by a person along a specified path. It uses the array EVALN, which contains a set of weights giving the desirability factor in various situations. If the person ultimately moves, then he will move along the path having the highest associated desirability.

NDRFIR determines whether a person should continue firing at a target, based on the firer's assessment of the target's physical status. Currently, firing will continue until the target is perceived as being dead or destroyed.

ODESIR computes the desirability of observing and finds the most desirable direction for observation. It uses the weights in EVALN. If the person chooses to observe, then he will observe in the most desirable direction.

TRGLST generates a list of targets at which a person may fire. Line-of-sight conditions are determined dynamically -- i.e., as they are needed, and from the geometry of the current situation.

SDESIR computes the desirability of surrender for a person, given his suppression level (see subsection 2.3.3). It uses the array EVALN.

The reader will note that the decision rules described above do not decide the issue of relative desirability of movement, firing, or observation. First and foremost, a person will try to follow his plans. If that is not

possible, and if logical constraints would allow the person to perform only one of these types of activities, then he does it. Otherwise, the order of desirability (most to least) is as follows: moving, firing, observing, and surrendering. The logical constraints on firing are basically common sense: the firer must have a suitable (e.g., within range) target; he must have ammunition; and he must not be suppressed in firing. The principal constraints on movement are that the mover is not suppressed in movement. Observation is permitted unless the person is suppressed in observation. Surrender can occur if higher priority activities cannot be done and if there is someone to surrender to. The ultimate possibility is that the person does nothing, finding no activity desirable.

2.3.6 Representation of Communications

After the physical activities of all persons have been simulated in a time step, communications between persons are processed. These communications are represented by lists of messages transmitted on communication nets. Each communication net is described by a COMNET record, containing the following types of information: its type (phone, CB radio type 1, CB radio type 2, voice, visual signal, alarm or siren, or loudspeaker); a reference to the list of persons that are members of the net, capable of transmitting and/or receiving; and a reference to a list of messages (if any) currently

on the net. Each message is described in a MESSAGE record, containing the following type of information: reference to the source person or sensor; a reference to the subject of the message; a specification of the attribute of the subject described in the message; the description of that attribute; and the time at which the message was generated, in seconds.

Communication by a person is automatic during the simulation. It takes place concurrently with any activity of the person, so long as he is not dead or captured. A person with information to transmit places that information on every communication net of which he is a member. Similarly, a person receives information from all communication nets he can monitor; i.e., he does not select a net to listen in on. The number of messages a person can transmit and receive during a time period is limited by input parameters (HFLIMS). The rate of flow of information among persons is also affected by the expected delays times (COMDLA) associated with each type of net.

The messages a person will transmit are of the following types: observations that differ with his previous perceptions; messages he has intercepted by listening in on his opponent's nets; and orders to his subordinates (if any).

There are limitations on the number of messages a person may receive, distinguishing four types of messages: orders to the receiving person; orders to other persons; observations of other (friendly) persons; and opponent's messages intercepted.

Each net has an associated list of messages. For each person, the logic finds the nets with which he is associated, searches the associated list of messages, and ignores the following: messages which are not yet available due to communication delay; messages the person sent himself; messages about the potential recipient, unless they are orders from his leader; and messages that match his current perceptions. The remaining messages on such lists are ordered in importance; then, they are merged with the recipient's own observations. These will then be used to update his perceptions.

2.4 Overview of Input Requirements

This section provides an overview of the input requirements of the Fixed Site Neutralization Model and its two preprocessors. These requirements are discussed in terms of general content, storage space needed, and input preparation. The organization of the discussion parallels the order of presentation of these three programs in other portions of this manual: the Plex Preprocessor is first, followed by the Data Preprocessor and the Fixed Site Neutralization model. All statements giving storage requirements should be interpreted as referring only to the original set of dimensions and code provided; appropriate adjustments should be made for any redimensioning or addition of new arrays. The presentation assumes, for the sake of completeness, that the user is "starting from scratch" and intends to exercise the programs for a new site and a new scenario. To properly estimate the magnitude of

the data-gathering effort required of the user, the reader should make allowances for the availability of portions or all of the data bases employed by previous users of the system.

User inputs to the Plex Preprocessor total roughly 500 to 1200 records, with the exact amount depending on the complexity of the site to be described and the composition of the forces prescribed by the scenario. (Exhibit 2-11 displays selected maximums for the numbers of entities and features that would be represented in the most complex case allowed by current dimensions in the programs.) These records are contained in two files, RECDESCRIP and RFDATA. The former contains complete descriptions of the site and the force on it at the beginning of the simulation; the latter describes a response force which may possibly arrive during the simulation. Preparation of RECDESCRIP and RFDATA is discussed in section 4.1 of this manual.

The other input file to the Plex Preprocessor describes the plex data structure. This is the file RECDESCRIP, consisting of 450 records. It is provided to the user, who should not make any changes in it without the advice of a programmer.

There is but one file input to the Data Preprocessor. It is the file that supplies values for most of the arrays to be used by the Fixed Site Neutralization Model. The bulk of it is performance data for the representation of people and mechanical devices. The major types of information

EXHIBIT 2-11: SITE AND FORCE COMPLEXITY PROVIDED FOR IN THE CURRENT DIMENSIONS	
SITE-RELATED MAXIMUMS	FORCE-RELATED MAXIMUMS
4 Buildings 8 Floors 26 Barriers 56 Yards 36 Rooms ³ 8 Halls ³ 6 Roofs 16 Stairs 58 Doors ⁴ 48 Windows ⁴ 105 Sensors ¹ 16 Remote Controls (Activated Delays)	50 Persons ¹ 85 Weapons ¹ 105 Items of Equipment ¹ (e.g., Keys) 5 Vehicles ² 16 Communication Nets 6 Adversary Goals

¹These maximums are estimated because the program needs the corresponding record types for *two* purposes: to describe the physical objects and to keep track of perceptions of these physical objects. We are concerned only with the first of these purposes in this table, but must set aside records for the other purpose.

²Vehicles are not currently used. There are some provisions for them in logic. For details, see section 4.18 of the programmer's manual.

³ROOM records may be used to describe halls; HALL records may be used to describe rooms. Therefore, there may be as many as $36 + 8 = 44$ rooms and/or halls in the site description.

⁴These 58 physical doors would correspond to $58 \times 2 = 116$ DOOR records in the site description, because both sides of a door are distinguished in the representation. Similarly, 48 windows would correspond to $48 \times 2 = 96$ WINDOW records.

involved are summarized in exhibit 2-12. Currently, there are 2301 values that must be supplied by the user. Because these values are usually for arrays, a record may be conveniently used to input more than one of these values; therefore, this is equivalent to roughly 200 records. Preparation of the Data Preprocessor input file is discussed in section 5.1 of this manual.

EXHIBIT 2-12: MAJOR TYPES OF INFORMATION INPUT TO THE DATA PREPROCESSOR	
MAJOR TYPE OF INFORMATION	SPECIFIC EXAMPLE
Degradation Factors	Effectiveness of Bullet-Proof Vests
Delays	Expected Delay, Communication Net 1
Human Factors Limitations	Maximum Messages Processed Per Time Cycle
Planning Weights	Relative Desirabilities for Situations
Priorities	Priority of Observing an Enemy Weapon
Probabilities	Probabilities of Kill Given a Hit
Ranges	Maximum Effective Range for Grenades
Rates of Activity	Meters Traveled per Second of Normal Walking
Skill Levels	Running Speed of Person 2, Relative to the Norm
Suppression Thresholds	Limitation on Firing Activities

Only eight additional values (comprising 1 record) need be provided by the user in order to obtain a Fixed Site Neutralization Model run. These are the initial firing conditions and random number seeds in file SEEDS, whose preparation is fully explained in section 6.1 of this manual. All other input files have been prepared by the preprocessors; therefore, there are no additional requirements of the user.

2.5 Overview of System Outputs

Assuming that a set of error-free runs has been produced by the Fixed Site Neutralization Model and its two preprocessors, there will be three major outputs of interest to the user:¹

- (1) A "report file", on logical device number 6, showing the resolution of the conflict and a brief summary of the course of battle. The summary shows the locations, activities, and observations of all people on the site at the end of every time interval--whose duration is under user control. An excerpt from such a file is shown in exhibit 2-13.
- (2) The "final file", on logical device number 8, presents a snapshot of the site at the moment the simulation ends. Everything is described. Items of interest include, but are by no means limited to, the following: location of the SNM; physical statuses of the guards; condition of

¹Outputs primarily of interest to programmers are deemphasized in this section.

EXHIBIT 2-13: EXCERPT FROM REPORT FILE ON LOGICAL DEVICE NUMBER 6

REPORT AT TIME:	1.67								
PN 1 (RM 6)	OBSERVING								
PN 2 (RM 5)	OBSERVING								
PN 2 (RM 5)	CAPTURING	PN 8							
SEES: PN 4	PN 5	PN 6	PN 8						
PN 3 (RM 5)	MOVING TO RM 5								KILLED
PN 4 (RM 5)	OBSERVING								KILLED
PN 5 (DR 12)	FIRING AT PN 4								
SEES: PN 2	PN 3	PN 4	PN 6	PN 8					
PN 6 (DR 12)	FIRING AT PN 3								
SEES: PN 2	PN 3	PN 4	PN 5	PN 8					
PN 7 (DR 19)	MOVING TO RM 5								
PN 8 (RM 5)	SURRENDERING TO PN 2								

Abbreviations used: DR, Door; PN, Person; RM, Room.

the physical plant (walls, doors, etc); and ammunition on hand. Excerpts from such a file are shown in exhibit 2-14 on the following page. In part 1 of that exhibit we can read off the contents of rooms 1 through 6 and roof 1 through 2, the status of doors 1 through 3, etc. The force summary in part 2 is particularly interesting: the fraction of effective strength (FRAC) is computed for each force; a code is given for the situation (SITN) each force is in; and list references (#) are given to the members of each force. (Two of these lists are shown at the bottom of part 2.) This will give some idea of the detail available. Overall, this file will provide 600 to 1500 lines of output--depending upon the complexity of the site and the composition of the forces specified by the scenario.

- (3) The "changes file", on logical device number 6, presents a record of every state variable change made during the entire simulation, in the order that they were made. This file is rarely listed because it is voluminous (100 to 200 pages) and, therefore, costly to print. Nonetheless, a particular event in the "report file" may generate questions that can be answered by *selectively* reading from this file if it is saved on disc or tape. Excerpts from this file are shown in exhibit 2-15.

EXHIBIT 2-14: EXCERPTS FROM THE FINAL FILE ON LOGICAL DEVICE NUMBER 8 (Part 1 of 2)

ROOM	ICoord	YCoord	ZCoord	CONTENT	SENSOR	HEIGHT	ENVIRON	XLEN	YLEN	LOS	COVER	ACCESS	CONT IN
1	106.50	87.00	0.0			8 390		21.00	18.00		0.20		FL 1
2	120.00	93.00	0.0					6.00	6.00		0.20		FL 1
3	120.00	87.00	0.0			8 391		6.00	6.00		0.20		FL 1
4	120.00	81.00	0.0	8 535				6.00	6.00		0.20		FL 1
5	130.50	87.00	0.0	8 2361		8 393		15.00	18.00		0.20		FL 1
6	36.00	21.00	0.0	8 394		DR 13		12.00	6.00		0.20		FL 2

LIST MOD

390 WI 2 DR 9 DR 19

391 DR 20 DR 21

535 SN 6 SN 5 SN 4 SN 3 SN 2 SN 1

2361 PH 7 PH 3 PH 2 PH 8 PH 4 AD 2 EQ 9 EQ 10

393 DR 11 DR 22

394 PH 1 AD 1 AD 3

ROOM	ICoord	YCoord	ZCoord	CONTENT	SENSOR	HEIGHT	ENVIRON	XLEN	YLEN	LOS	PENT
1	117.00	87.00	3.60					42.00	18.00		5
2	36.00	21.00	3.60					12.00	6.00		5

DOOR	ICoord	YCoord	ZCoord	CONTENT	SENSORS	HEIGHT	ENVIRON	STATUS	LOCKBL	LOS	PENET	VISIBL	ACCESS	WIDTH
HEIGHT PORTAL CONT IN														
1	48.00	12.15	0.0			8 108		4	2		3	3	1	6.00
2.40 DR	2	BR	1											
2	48.00	11.85	0.0			YD 46		4	2		3	3	1	6.00
2.40 DR	1	BR	1											
3	191.85	90.00	0.0		SN 1	8 109		5	1		3	3		0.90

EXHIBIT 2-14: EXCERPTS FROM THE FINAL FILE ON LOGICAL DEVICE NUMBER 8 (Part 2 of 2)

RELATION	TYPE	PAR1	PAR2
1	12		
2	12		
3	3		
4	3		
5	3		
6	3		
7	3		
8	12		

GOAL	NOD	REGION	SUBORD	MIN REQ	CONTROL EQUIP	MEN REQ
1	RM	5		2.00		2

NODE-OF-	NODE	ID	PAR1	PAR2	PLAYER	NOISE
85	DR	20	1.67		PN 7	
86	RM	3	1.70		PN 7	
87	DR	21	1.72		PN 7	
88	DR	22	1.79		PN 7	

FORCE	TYPE	ALLEG	LEADER	CONTENT	PRAC	SITN	PLANS	SOPS
1	1	1	PN 1	# 477	0.50	5		
2	1	2	PN 5	# 478	1.00	4		
3	1	1	PN 9	# 525	1.00	1		
4	1	1	PN 1	# 647	1.00	4		
5	1	2	PN 5	# 652	1.00	4		
6	1	2		# 1118	1.00	3		
7	1	1		# 1947	0.67	4		

LIST	NOD							
477	PN	1	PN	2	PN	3	PN	4
478	PN	5	PN	6	PN	7	PN	8

EXHIBIT 2-15: EXCERPTS FROM THE CHANGES FILE ON LOGICAL DEVICE
NUMBER 6 (Part 1 of 3)

*** INITIALIZATIONS

TMIN = 0.0
NMSTAT = 1
FORCES(1, 1) = 0.0
FORCES(2, 1) = 0.0
FORCES(3, 1) = 0.0
FORCES(4, 1) = 4.00
FORCES(1, 2) = 0.0
FORCES(2, 2) = 0.0
FORCES(3, 2) = 0.0
FORCES(4, 2) = 4.00
IGOALS = -9999
RECORD GL 1 QUEUED ON LIST IGOALS

*** LEADER PLANNING

FIELD FRAC	OF FR	1	CHANGED FROM	1.0	TO	1.0
FIELD IDENT	OF PC	42	CHANGED FROM		TO FR	1
FIELD VIEW	OF PC	42	CHANGED FROM		TO FR	4
FIELD CONTENT	OF FR	4	CHANGED FROM		TO PN	1
FIELD CONTENT	OF FR	4	CHANGED FROM	PN 1	TO #	645
FIELD CONTENT	OF FR	4	CHANGED FROM	# 645	TO #	646
FIELD CONTENT	OF FR	4	CHANGED FROM	# 646	TO #	647
FIELD FRAC	OF FR	4	CHANGED FROM	-9999.0	TO	1.0
FIELD SITN	OF FR	4	CHANGED FROM	-9999	TO	1
FIELD FRAC	OF FR	2	CHANGED FROM	1.0	TO	1.0
FIELD SITN	OF FR	2	CHANGED FROM	2	TO	1
FIELD IDENT	OF PC	43	CHANGED FROM		TO FR	2
FIELD VIEW	OF PC	43	CHANGED FROM		TO FR	5

EXHIBIT 2-15: EXCERPTS FROM THE CHANGES FILE ON LOGICAL DEVICE
NUMBER 6 (Part 2 of 3)

NEW MESSAGE RECORD:

	SOURCE	SUBJECT	ATTRIB	TIME	CONTENT
87	PN 2	PN 6	2	1.33	3
RECORD MS 87 STACKED ON LIST LASMNT					
FORCES(3, 2)			=	0.0	
FORCES(1, 2)			=	1.00	
IPLAYR			=	1	
IPLAYR			=	2	
IPLAYR			=	5	
IPLAYR			=	7	
IPLAYR			=	8	

*** DETECT OBJECTS

ISIDE	=	1
IPLAYR	=	1
NFSNLT	=	-9999
IPLAYR	=	2
ISIDE	=	2
IPLAYR	=	5
IPLAYR	=	6
IPLAYR	=	7
IPLAYR	=	8

*** PROCESS OBSVNS

ISIDE	=	1
IPLAYR	=	1
FIELD UPDATED	OF PC	45 CHANGED FROM 1.2 TO 1.3
FIELD UPDATED	OF PC	46 CHANGED FROM 1.2 TO 1.3
FIELD UPDATED	OF PC	47 CHANGED FROM 1.2 TO 1.3
FIELD UPDATED	OF PC	48 CHANGED FROM 1.2 TO 1.3
FIELD UPDATED	OF PC	49 CHANGED FROM 1.2 TO 1.3
FIELD UPDATED	OF PC	51 CHANGED FROM 1.2 TO 1.3

NEW MESSAGE RECORD:

	SOURCE	SUBJECT	ATTRIB	TIME	CONTENT
86	PN 1	PN 1	9	1.33	# 1093
FIELD UPDATED OF PC 1 CHANGED FROM 1.2 TO 1.3					
FIELD MSG OF CN 1 CHANGED FROM # 1131 TO # 1092					
IOBSV (1)			=	-9999	
IPLAYR			=	2	
FIELD UPDATED			OF PC	62 CHANGED FROM 1.2 TO 1.3	

NEW MESSAGE RECORD:

	SOURCE	SUBJECT	ATTRIB	TIME	CONTENT
85	PN 2	PN 2	6	1.33	4

EXHIBIT 2-15: EXCERPTS FROM THE CHANGES FILE ON LOGICAL DEVICE
NUMBER 6 (Part 3 of 3)

*** NEW TIMESLICE

THIN = 2.5)

*** CHECK FOR RESPONSE FORCE ARRIVAL
RESPONSE FORCE DOES NOT ARRIVE

*** CHECK FOR END
SIMULATION CONTINUES

*** LEADER PLANNING

FIELD FRAC	OF FR	1	CHANGED FROM	0.5	TO	0.5
FIELD FRAC	OF FR	6	CHANGED FROM	0.5	TO	0.5
FIELD FRAC	OF FR	4	CHANGED FROM	1.0	TO	1.0
FIELD FRAC	OF FR	2	CHANGED FROM	0.5	TO	0.5
FIELD FRAC	OF FR	7	CHANGED FROM	0.7	TO	0.7
FIELD FRAC	OF FR	5	CHANGED FROM	1.0	TO	1.0

*** PERSON DECISIONS

ISIDE	=	1				
IPLAYR	=	1				
FIELD ACTION	OF PN	1	CHANGED FROM		TO AC	1
IPLAYR	=	2				

NEW ACTION RECORD:

TYPE	PAR1	PAR2	PAR3	MODIFIER	DOER	
154 3	0.25				PN 2	
FIELD ACTION	OF PN	2	CHANGED FROM		TO AC	154
ISIDE	=	2				
IPLAYR	=	5				

NEW ACTION RECORD:

TYPE	PAR1	PAR2	PAR3	MODIFIER	DOER	
156 3	3.14				PN 5	
FIELD ACTION	OF PN	5	CHANGED FROM		TO AC	156
IPLAYR	=	8				

NEW ACTION RECORD:

TYPE	PAR1	PAR2	PAR3	MODIFIER	DOER	
158 3	1.57				PN 8	
FIELD ACTION	OF PN	8	CHANGED FROM		TO AC	158

2.6 Resource Requirements for the Use of the System

This section provides an overview of the resources required to use the Fixed Site Neutralization Model and its preprocessors. The requirements are addressed in the following order: time of personnel to prepare the data and run the programs, computer funds, and storage.

To prepare the site and response force descriptions (i.e., the RECDATA and RFDATA files input to the Plex Preprocessor) would take someone familiar with the system about one man-week. The gathering of values for the data variables would be a considerable undertaking, if starting from scratch. Fortunately, reasonable values for most of these variables are already known, and the user is advised to avail himself of them. The only other significant expenditure of personnel time would be in the analysis of the outputs, and this would vary with the nature of the study being performed. Time spent at the terminal submitting a run of any of the programs is minimal and on the order of one to five minutes.

The Plex Preprocessor is rather inexpensive to run. Depending on the nature of the scenario and the complexity of the site, a typical run might cost \$1.00 to \$2.00, at commercial rates on the University of Michigan machine. The Data Preprocessor is even less, typical runs cost less than \$1.00. Fixed Site Neutralization Model runs have cost somewhere between \$7.00 and \$65.00, depending upon the amount of simulated time it takes to resolve the situation presented in the scenario.

Storage requirements for the system are as follows on the AMDAHL 470V/6 machine at the University of Michigan: the Plex Preprocessor runs in approximately $75K_{10}$ 32-bit words; the Data Preprocessor, $6.6K_{10}$; and the Fixed Site Neutralization Model, $98K_{10}$.

3.0 PLANNING THE RUNS

To plan a study that would involve the use of the Fixed Site Neutralization Model and its preprocessors, the user should consider many factors: the study objective(s); the scenario(s) to be used; the availability of data and other resources, such as calendar time, computing time and personnel; and the special characteristics of and the interrelationships among the programs to be used. While it is clearly not feasible to anticipate all of the various possible combinations of studies, scenarios, and sets of circumstances in which these programs could be used, it is nonetheless possible to offer the first-time user some general guidance for the planning of his runs. This chapter offers such guidance, stressing the special characteristics, options, and sometimes subtle interrelationships that exist among these programs and their associated data bases. It is hoped that this material will promote an efficient use of the programs, so that: the planned number of runs more closely resembles the actual; and the anticipated timing of data base developments and model uses proves to be realistic.

The chapter is organized into three parts, each corresponding to a program: section 3.1 offers guidance on the planning of Plex Preprocessor runs; section 3.2, Data Preprocessor runs; and section 3.3, Fixed Site Neutralization Model runs.

3.1 Planning Plex Preprocessor Runs

The Plex Preprocessor is a rather inexpensive program to run. Depending on the scenario and the complexity of the site, a typical run might be \$1.00 to \$2.00, at commercial rates on the University of Michigan machine. Anywhere from six to 12 runs might be expected to set up the plex structure for a new site and scenario--although very experienced personnel might well do it in less. If only the scenario is new and the site remains unchanged, two to three runs would probably be needed. If, as a result of inspecting FSNM outputs, it is desired to change only one or two items from a previous run, then only one run would typically be required.

For reasons of efficiency, some procedures are recommended in building the RECDSCRIP file. One would start, of course, with a good scale drawing of the site. The order in which the the various types of records are input is not prescribed; however, the following order has several practical advantages. First, do the records describing the exterior and gross features of the site: SITE, BUILDING, FLOOR, BARRIER records, as well as the LIST NODE (LIST) records they reference. Second, do the DOOR and WINDOW records, and their associated LIST records. (Some prefer to do the exterior DOOR and WINDOW records here, saving the corresponding interior records for later.) Now the YARD records can be done, and the neighbors referred to will be easy to determine. Then one can concentrate on the interior details of the site: SENSOR, ROOM, HALL, and STAIR records, for example, and their associated LIST records. Because it is much more easy to add records to the plex data structure than to subtract from it, a

simple version of the interior may suffice. For example, some of the small offices might be consolidated into larger ones--with judgment reserved as to whether the loss of detail could affect the results. (Where the action takes place--as determined by an FSNM run--is where the detail should be; other detail may be superfluous.) It is also easier to modify the plex data structure later, for differing scenarios on the same site, if the site is described first as suggested above.

The plex data structure is then completed with the scenario-dependent data. This involves the PERSON, FORCE, COMNET, ACTION, LOCATION, EQUIPMENT, WEAPON, and GOAL NODE records, as well as the associated list records. The PERCEPT records, as well as the records referred to in their CURRENT VIEW fields, could be saved till last or omitted. At the end, the preparer of the data base should check that all record types he wishes to view in output have at least one record in the input, so that the program has a format to refer to. For example, MESSAGE and NODE-OF-IMPACT records need not be input at all, but a dummy of each type could give the necessary format information.

Throughout the RECDESCRIP preparation, decisions will be made that will affect records that have not yet been prepared. Liberal use of the provision for comment cards within this file will be a handy way to keep track of all that. In general, as the various record types are numbered for identification, one should use a consecutive numbering scheme starting with 1. This takes full advantage of the error-checking logic of the PPP; see section 6.1 for details. An incremental approach may be adopted to the building of this file. At any stage mentioned above, one can stop, run the PPP,

and use its advisory messages in output to determine what remains to be done. This is useful to discover oversights -- i.e., errors of omission.

When all files to the PPP have been prepared, and an evidently satisfactory run of that preprocessor has been obtained, the study team may find it advantageous to have someone review the input files for consistency with the scenario. The idea is that PPP runs are inexpensive, relative to the FSNM runs and their associated analysis. An inconsistency with the scenario that inadvertently passes this stage may not become apparent until much computer and study personnel time have been wasted. For a complicated site and/or scenario, although reviewing the data structure may look like an overwhelming task, aids such as maps that the PPP outputs allow one to check many complex relationships at a glance.

3.2 Planning Data Preprocessor Runs

The Data Preprocessor is an inexpensive program to run. While the exact charge depends on the nature of the scenario, typical runs cost less than \$1.00, at commercial rates on the University of Michigan machine. One run is always necessary. The number of reruns to plan for depend upon the types of sensitivity analyses to be performed. Ordinarily, one need not plan for any more runs to allow for errors--provided strict precautions are taken, as described below.

The Data Preprocessor is a "dumb" program; essentially, it cannot detect errors in its inputs. If the inputs are thoroughly checked, one

can proceed with confidence, because the program has been thoroughly debugged.

Prior to the run, the header records should be compared to the /DATA/ common, to be sure the right order is used and that the dimensions match. It is important that the format on the header records match that actually used in the data. The program has no way to check that this occurs. (The study team may wish to reprogram the DPP so that it echos back the data, so that inputs and outputs can be compared; this would be a minor change.)

After the run, the echo of the header records should be compared to the input. The field pointer associated with the ENDBLK card should be 2103 (the size of common /DATA/). No other outputs are currently produced that could be used to check for errors.

The planner should keep in mind that these efforts are primarily to avoid useless runs of the relatively costly FSNM, and associated expenditures of analysis time. Considered by themselves, the DPP runs are of deceptively little consequence on resources.

3.3 Planning Fixed Site Neutralization Model Runs

Depending upon the complexity of the site and scenario being played, the planner can expect runs of the Fixed Site Neutralization Model to cost somewhere between \$7.00 and \$65.00. Of the three programs documented in this manual, this is the most costly in terms of computer time. Fortunately, there are relatively few ways a FSNM run can be unproductive--*assuming* that its input files have been properly prepared to correspond to the intended scenario. This makes the checks and precautionary

procedures mentioned in the previous two sections all-important in determining the efficient use of this model.

For every site/scenario combination, one can plan on one FSNM run. Allowance for sensitivity analyses runs, in accordance with the needs of the study, can also be made on the same basis. One sensitivity run should be done at an early stage in any extended study to check that the time interval specification is not having an appreciable effect on the output results of interest. For example, suppose that the first apparently productive FSNM run is produced with a time step of 15 seconds, and that eight parametric runs are planned introducing slight variations. Before the eight runs, it would be prudent to see that the time interval selected was not appreciably influencing the results. Suppose a six-second time interval is chosen and the FSNM is then run with all other inputs as in the first run. If no appreciable differences occur, one can proceed with the 15-second time interval on the remaining runs, with some assurance that the time interval played is unimportant. But if appreciable differences occur when the comparison of the six-second and 15-second time interval runs are made, one could conclude that the duration of this interval should probably not be 15 seconds in the remaining runs. The appropriate length would have to be determined by subsequent runs, trying other lengths less than 15 seconds. Thus, in this example, eight useless runs of the FSNM are precluded by adherence to the suggested procedure. When the parametric runs are made, they will be with a demonstrably reasonable time-step duration.

It is also important to go over the early runs of the model to check if any actions are taking place in areas (or regions) that have been simplified in the site representation. If so, the plex data structure should be ammended to include hitherto suppressed details in the description of that area, before any more runs are made. A small office, for example, may provide an attacker with just the place he needs to take cover at a certain time; any simplified version of the site ommitting it may materially influence the outcome.

BLANK PAGE

4.0 USING THE PLEX PREPROCESSOR

This chapter covers three subjects related to the use of the Plex Preprocessor: preparation of inputs, treated in section 4.1; running the program, section 4.2; and responding to outputs, section 4.3. The treatment of these subjects is generally at a nontechnical level. The emphasis is on making the user (or potential user) comfortable with the tasks and cognizant of the location of all materials necessary for the accomplishment of those tasks. Suitable cross-references to the programmer's manual are given at points, to avoid unnecessary duplication of material.

4.1 Preparing Plex Preprocessor Inputs

The bulk of the data input to the Fixed Site Neutralization Model is descriptive of the site and scenario, and is stored in a plex data structure. Consequently, these data are processed by the Plex Preprocessor, and the proper preparation of inputs to that program is of great importance. Three input files are needed: RECDESCRIP, which describes the plex data structure and the various types of records in it; RECDATA, which describes the initial conditions at the site; and RFDATA, which describes the response force to arrive at the site (after a scenario-dependent delay). We will discuss the preparation of these three files in turn.

File RECDESCRIP describes the plex data structure. Ordinarily, it is not prepared by the user; instead, the current version of the plex data structure is used, so that the user need only refer to the already existing file in his run command. The user should understand that changes to this file are *structural*, and have implications for the *logic* of the program. Therefore, it is recommended that a programmer be consulted if any changes are desired in this file. The reader is referred to other sections of the documentation for further details, as shown in exhibit 4-1.

EXHIBIT 4-1: DOCUMENTATION OF FILE RECDESCRIP	
PORTION OF PROGRAMMER'S MANUAL	CONTENT OF POTENTIAL INTEREST TO USERS
Section 2.1	Description of Plex Preprocessor logic, including the processing of RECDESCRIP
Chapter 3.0	General readings on plex data structures
Section 3.2.1	Format of RECDESCRIP records
Chapter 4.0	Definitions of the contents of all plex record types, as provided by the current structure in RECDESCRIP
Section 7.2	Source listings for the Plex Preprocessor
Section 8.1	Current listing of the RECDESCRIP file

File RECDATA is prepared by the user. The person doing it for the first time would do well to have a plan of attack such as the following:

- (1) Become aware of the type of information that will be asked of you. Do this by reading chapter 4 of the programmer's manual, where the various record types are explained. It is *not* essential to know about plex data structures, or how they are processed to do this. Your task is to supply the information that will be put into the fields of the records. Do not concern yourself with *how* it is put in at this time.
- (2) Get a general picture of what the RECDATA file will look like when it is done by consulting exhibit 8-2 of the programmer's manual.
- (3) Obtain the necessary information about the site and the scenario. The best site information summary would be a map, laid out on quadrille paper, so that points (nodes) of interest can be located in a consistent coordinate system.
- (4) Plan ahead, using the suggestions from section 3.1 of the user's manual. This gives you the order in which the various types of records can be efficiently prepared. Stick to it.
- (5) Following the sequence specified in (4), prepare one type of record at a time. Refer to the appropriate section of chapter 4 of the programmer's manual for the definitions of the items of information (fields) that go into each record. Format the information in accordance with section 3.2.2 of the programmer's manual. The task is manageable if you do it one record type at a time, in the suggested order.

- (6) Run the Plex Preprocessor, using your inputs, following the instructions in section 4.2 of the user's manual.
- (7) Use any error or advisory messages to determine the revisions needed. Remember, the map that is output is invaluable in spotting things that are out of place, left out, etc. Return to previous steps as necessary.

The preparation of RFDATA is also done by the user. The file consists of a limited set of record types, and each of these record types is also used in RECDATA. Therefore, a *parallel* procedure to the one just detailed for RECDATA can be followed:

- (1) Same as with RECDATA. Chapter 4 readings apply to *both* files.
- (2) Same idea: get a general picture of what the RFDATA file will look like when it is done by consulting exhibit 8-3 of the programmer's manual.
- (3) Obtain the necessary information about the response force.
- (4) The order in which to consider the records is the same as that used for RECDATA. The only difference is that many of the record types will not be needed for RFDATA.
- (5) Same as with RECDATA.
- (6) Same as with RECDATA.
- (7) Same as with RECDATA. You can debug both files at the same time.

4.2 *Running the Plex Preprocessor*

The run commands for the Plex Preprocessor are as follows:

```
$CREATE PLEXOBJ SIZE=44P  
$RUN PLEXPOBJ+UTIL.LIB+BLOCKDATAOBJ 3=RECDESCRIP 4=RECDATA 5=RFDATA  
6=PLEXOBJ 7=*SINK* 8=OUT 9=-MAP
```

PLEXOBJ is the output file for the binary plex data structure. The first command creates a permanent file to store it, so that it may be used later as input to the Fixed Site Neutralization Model. PLEXPOBJ is the object file for the Plex Preprocessor; this is the logic of the main program and its subprograms in machine-readable form. UTIL.LIB is a library of utility programs -- again, an object file. BLOCKDATAOBJ is the object file for the BLOCKDATA subroutine, which specifies the dimensions of the various arrays used by the program. The numbers referred to in the RUN command -- 3 through 9 -- are logical device numbers. Their assignments to the various files are accomplished with the '=' signs. RECDESCRIP is the file describing the plex data structure (use exhibit 4-1 to locate information on this file). RECDATA is the initial description of the site and the scenario; its preparation by the user is discussed in section 4-1 of this manual. RFDATA is the description of the response force; its preparation by the user is discussed in section 4-1 of this manual. *SINK* is a reference to the printer, where error messages may be written: in interactive mode, this will be the terminal; in batch mode, the line printer. -OUT is a temporary file to contain the listing of the plex structure. To get a listing of this file, follow the run command with: \$LIST -OUT. Note that such a listing is necessary, because the Plex Preprocessor may rearrange the user-supplied numbers identifying his LIST records.

4.3 *Responding to Plex Preprocessor Outputs*

The Plex Preprocessor has four output files: the binary file on logical device number 6; the error and advisory messages, on logical device number 7; the plex structure listing, with new LIST record numbers, on logical device number 8; and the map, on logical device number 9. User responses to the contents of these files are discussed below.

The file on logical device number 6 is unformatted -- i.e., it has no printable characters. Therefore, the contents of this file are not inspected. To ascertain if the file satisfactorily encodes the desired data, one inspects the error messages on logical device number 7, the plex data structure shown on logical device number 8, and the map on logical device number 9. If no difficulties are ascertained after such an inspection, the file PLEXOBJ may be input to the Fixed Site Neutralization Model with confidence. If one or more serious difficulties are detected, PLEXOBJ is of no use and may be destroyed or emptied. (The corresponding MTS commands would be \$DESTROY PLEXOBJ or \$EMPTY PLEXOBJ, and the confirming command used to respond to subsequent prompting are the letters OK.) After the deficiencies in the Plex Preprocessor input have been remedied, the program would be rerun.

The error and advisory messages produced on logical device number 7 are fully discussed in section 6.1 of the programmer's manual. Note that errors numbered 8 through 16, inclusive, indicate deficiencies in the input which can be remedied by the user. The other messages indicate potential bugs in the program; should one of them occur, the assistance of a programmer should be obtained.

The file on logical device number 8 is the intelligible version of the binary file on logical device number 6. This should be carefully inspected. With the exception of list renumbering, the plex data structure output by the program should be compared side-by-side with the intended version. Misalignment of data within the fields should be watched for, as well as typographical or keypunching errors. The task may seem tedious, but this is a case where "an ounce of prevention is worth a pound of cure."

What is difficult to spot in the perusal of the file on logical device number 7 are errors of omission (e.g., a missing window), errors of location (e.g., two yards in the same place), and misspecified relationships (e.g., two supposed "neighbors" that have a wall between them). These errors are most easily detected by comparing the map output on logical device number 8 with the map from which the user prepared the data inputs. One should systematically search for all these types of errors, regardless of the presence or absence of error and/or advisory messages on logical device number 7. It is far less costly to discover such errors at this stage, causing the Plex Preprocessor to be rerun, than it is to discover them later, after one or more Fixed Site Neutralization Model runs are invalidated.

BLANK PAGE

5.0 USING THE DATA PREPROCESSOR

This chapter covers three subjects related to the use of the Data Preprocessor: preparation of inputs, treated in section 5.1; running the program, section 5.2; and responding to outputs, section 5.3. The treatment of these subjects is generally at a nontechnical level. The emphasis is on making the user (or potential user) comfortable with the tasks and cognizant of the location of all materials necessary for the accomplishment of those tasks. Suitable cross-references to the programmer's manual are given, to avoid unnecessary duplication of material.

5.1 *Preparing Data Preprocessor Inputs*

Virtually all data input to the Fixed Site Neutralization Model in array form is processed by the Data Preprocessor. (The bulk of it is performance data.) There is but one input file needed. A complete example of an input file is shown in exhibit 8-4 of the programmer's manual. The arrays are to be described and then supplied with data in the exact *order* shown in that exhibit. The formats of the header records are as given in exhibit 2-3 of the programmer's manual. The formats of all other records, supplying the actual data, may be set at the user's convenience; the only requirement is that these formats be properly described, using the usual FORTRAN conventions, on the associated header records.

CONTENTS

	<u>Page</u>	
1.0 INTRODUCTION	11	1/A11
1.1 Purpose of the Manual	12	1/A12
1.2 Organization of the Manual	12	1/A12
1.3 Conventions Used in the Manual	13	1/A13
2.0 OVERVIEW	17	1/B3
2.1 Initial Overview of the System of Programs	18	1/B4
2.2 Initial Overview of the Fixed Site Neutralization Model	20	1/B6
2.3 Extended Overview of the Fixed Site Neutralization Model	28	1/B14
2.3.1 Site Representation	28	1/B14
2.3.2 Entities Represented	32	1/C4
2.3.3 Activities Represented	38	1/C10
2.3.4 Representation of Perceptual Processes	44	1/D4
2.3.5 Representation of Decision Processes	53	1/D14
2.3.6 Representation of Communications	60	1/E7
2.4 Overview of Input Requirements	62	1/E9
2.5 Overview of System Outputs	66	
2.6 Resource Requirements for the Use of the System	74	1/E13
		1/F11
3.0 PLANNING THE RUNS	75	1/F12
3.1 Planning Plex Preprocessor Runs	76	1/F13
3.2 Planning Data Preprocessor Runs	78	1/G1
3.3 Planning Fixed Site Neutralization Model Runs	79	1/G2
4.0 USING THE PLEX PREPROCESSOR	83	1/G6
4.1 Preparing Plex Preprocessor Inputs	83	1/G6
4.2 Running the Plex Preprocessor	87	1/G10
4.3 Responding to Plex Preprocessor Outputs	88	1/G11
5.0 USING THE DATA PREPROCESSOR	91	1/G14
5.1 Preparing Data Preprocessor Inputs	91	1/G14
5.2 Running the Data Preprocessor	92	2/A2
5.3 Responding to Data Preprocessor Outputs	93	2/A3
6.0 USING THE FIXED SITE NEUTRALIZATION MODEL	95	2/A5
6.1 Preparing the Fixed Site Neutralization Model Inputs	95	2/A5
6.2 Running the Fixed Site Neutralization Model	96	2/A6
6.3 Responding to Fixed Site Neutralization Model Outputs	97	2/A7

For the definitions of the variables involved, the user should refer to section 5.1 of the programmer's manual. The subscripts used to index the arrays are as defined in section 5.2 of the programmer's manual; the meanings of the various positions within the arrays are determined by this information.

5.2 Running the Data Preprocessor

The run command for the Data Preprocessor is as follows:

```
$CREATE DATAOBJ SIZE=4P  
$RUN DATAPPOBJ 5=DATA 6=DATAOBJ 7=*SINK*
```

The first line creates a permanent file, DATAOBJ, which will receive the binary outputs; assuming a successful Data Preprocessor run, it will be ready for direct input to the Fixed Site Neutralization Model. \$RUN is the MTS command which will cause the program to execute. The \$ is an optional character, if the command is given from the terminal. DATAPPOBJ is the permanent file containing the Data Preprocessor logic in machine-readable form -- i.e., its object file. The numerals 5, 6, and 7 are logical device numbers; the '=' following each of them indicates the assignment of the various files, consistent with Data Preprocessor input/output logic. DATA is the input data file whose preparation is discussed in section 5.1 of this manual. *SINK* is a reference to the printer, directing the echo of the header records to the terminal (in interactive mode) or the line printer (in batch mode).

5.3 *Responding to Data Preprocessor Outputs*

There are two output files of the Data Preprocessor. The one on logical device number 6 is unformatted -- i.e., it has no printable characters. Therefore, the contents of this file are not inspected. To ascertain whether the contents of this file are appropriate for input to the Fixed Site Neutralization Model, one should carefully follow *all* suggestions in section 3.2 of this manual. (One of the suggestions is to peruse the echo of the header records, located on logical device number 7.) Assuming that the various checks recommended there have been performed, the file DATAOBJ may be used with confidence.

BLANK PAGE

6.0 USING THE FIXED SITE NEUTRALIZATION MODEL

This chapter covers three subjects related to the use of the Fixed Site Neutralization Model: preparation of inputs, treated in section 4.1; running the program, section 4.2; and responding to outputs, section 4.3. The treatment of these subjects is generally at a nontechnical level. The emphasis is on making the user (or potential user) comfortable with these tasks and cognizant of the location of all materials necessary for the accomplishment of those tasks. Suitable cross-references to the programmer's manual are given at points, to avoid unnecessary duplication of material.

6.1 *Preparing Fixed Site Neutralization Model Inputs*

The Fixed Site Neutralization Model requires three input files: PLEXOBJ, prepared by the Plex Preprocessor; DATAOBJ, prepared by the Data Preprocessor; and SEEDS, a file whose contents and preparation are discussed below.

The SEEDS file has one record, with the following FORTRAN format: 2I1, 6I10. The integer values in columns 1 and 2 are IFCOND(1) and IFCOND(2), which refer to the site force and the adversary force, respectively. These are codes for the initial firing conditions for each side: 1 means do not fire; 2 means fire only if fired upon; and 3 means fire when a target is

available. The next six fields should contain different, positive, odd integers that will serve as random number seeds. The following values are suggested: 9876543, 9875309, 9874075, 9872841, 9871607, and 9870373. The user is cautioned that other random number seeds may be undesirable. Any contemplated changes should be checked with a programmer who is familiar with the pseudo-random number generator used.

6.2 Running the Fixed Site Neutralization Model

The command to run the FSNM is as shown below.

```
$RUN OBJ+UTIL.LIB+BLOCKDATAOBJ+NAAS:SSP 4=DATAOBJ 5=PLEXOBJ 6=*DUMMY*  
7=*SINK* 8=*DUMMY* 9=SEEDS
```

The \$RUN is the MTS (Michigan Terminal System) command causing the execution of the program. If the user is on the terminal, the \$ is optional. OBJ is the object file for the Fixed Site Neutralization Model. UTIL.LIB is the object file for the utility routines used. NAAS:SSP is the Scientific Subroutine Package, which contains the object file for the pseudo-random number routines. The numbers 4 through 9 are logical device numbers; the '=' following them indicates that assignments are being made. DATAOBJ is the binary file produced by the Data Preprocessor. PLEXOBJ is the binary file produced by the Plex Preprocessor. Each use of *DUMMY* indicates the user's desire to suppress the files that otherwise might be saved by and/or printed from the indicated logical device numbers. The file written to

logical device number 6 is ordinarily quite voluminous: it contains a record of every state variable change produced by the simulation, and is ordinarily not used except for debugging purposes. Consequently, it is economical in most cases to use the **DUMMY** option. (To save the file, insert the following command before the RUN command: *\$CREATE -CHANGES SIZE=100P*. Then change *6=*DUMMY** to read *6=-CHANGES* in the RUN command itself. To print the saved file, follow the RUN command with this command: *\$LIST -CHANGES*. The file involved in these suggested commands is a temporary file, due to the relatively large amount of storage occupied.) **SINK** is a reference to the printer, directing the error messages and a summary of the simulation to the terminal (in interactive mode) or the line printer (in batch mode). The file assigned to logical device number 8 is the plex data structure at the end of the simulation; to obtain it, replace *8=*DUMMY** with *8=-OUT* in the RUN command, and then write *\$LIST -OUT* as the next command. SEEDS is the file whose contents and preparation is discussed in section 4.1 of this manual.

6.3 Responding to Fixed Site Neutralization Model Outputs

The Fixed Site Neutralization Model produces three output files. Assuming that there is a successful run, there will be no error messages or advisory messages of consequence on logical device number 7, and the summary report printed by the same device may be all that is of interest.

(For the information content of that report, see section 2.6 of this manual; the example output in exhibit 9-6 of the programmer's manual might also be of interest.) The next level of detail that can be obtained is to inspect the contents of the output file on logical device number 8; one can think of this material as a "snapshot" of everything at the final moment of the simulation. If details concerning the course of events that took place are of interest, then the file written on logical device number 6 should be printed and perused. (The reader is reminded that the file on logical device number 6 is voluminous and relatively costly to print out.)

An unsuccessful run will be marked by error messages and/or advisory messages that are deemed consequential. These messages are explained in detail in chapter 6.1 of the programmer's manual. The user will typically *not* be able to investigate these messages without the help of an experienced programmer. The most important thing for the user to realize, if he is at a terminal, is that the files on logical devices numbered 6 and 8 may be of interest to the programmer who investigates. Consequently, if it is possible to do so, the contents of these files should be saved in a permanent file. (It will not be possible to do this if the otherwise economical *DUMMY* option, explained in section 6.2, was used in connection with these files. In such a case, the programmer may decide to perform the run all over again, just to get those files.) This may be done with the RENAME command, in the following manner: \$RENAME -CHANGES AS CHANGES and \$RENAME -OUT AS OUT.

That is, the removal of a minus prefix from a file name changes it from a temporary file into a permanent file. These files could then be read selectively by a programmer who is making a later investigation.

BLANK PAGE

INDEX

This index includes all programs, subroutines, functions, input files, and output files covered in the user's manual. The order of presentation is alphabetical, with the decimal numbers considered to be at the end of the alphabet.

ARRIVL subroutine	14
BLOCKDATA subroutine	87
BLOCKDATAOBJ file	87
Data Preprocessor - Main Program	65, 78 to 79, 91 to 93
DATAOBJ file	92, 95 to 96
DATAPPOBJ file	92
/DATAV/	79
EFFTIM function	57
FDESIR subroutine	59
Fixed Site Neutralization Model - Main Program	20 to 40, 79 to 81
	95 to 99
LDRPLN subroutine	14, 56
LIST function	88
MDESIR subroutine	59
NDRFIR function	59
NEWPCP subroutine	14
OBJ file	96
ODESIR subroutine	59
PLEXOBJ file	87 to 88, 95 to 96
PLEXPOBJ file	87
Plex Preprocessor - Main Program	76 to 78, 83 to 89
RECDATA file	74, 83, 85 to 87
RECDESCRIP file	63, 76 to 77, 83 to 84, 87
RFDATA file	74, 83, 86 to 87
SDESIR subroutine	59
SEEDS file	66, 95, 97
TRGLST subroutine	59
UTIL.LIB file	87, 96

DISTRIBUTION:

U.S. Nuclear Regulatory Commission (10)
MS 1130SS
Washington, DC 20555
Attn: R. Robinson

Los Alamos Scientific Laboratory
Attn: G. R. Keepin, R. A. Gore, E. P. Schlönka, D. G. Rose
Los Alamos, NM 87544

Allied-General Nuclear Services
Attn: G. Molen
P.O. Box 847
Barnwell, SC 29812

Lawrence Livermore Laboratory
University of California
P.O. Box 808
Attn: A. J. Poggio
Livermore, CA 94550

Pritsker and Associates, Inc.
P.O. Box 2413
Attn: F. H. Grant
West Lafayette, In 47906

Union Carbide Corporation
Nuclear Division
Bldg. 7601
Attn: D. Swindle
Oak Ridge, TN 37830

400	C. Winter
1000	G. A. Fowler
1213	V. E. Gibbs
1230	W. L. Stevens, Attn: R. E. Smith, 1233
1700	W. C. Myre
1710	V. E. Blake, Attn: M. R. Madsen, J. W. Kane
1716	R. L. Wilde, Attn: B. D. Link, 1716
1730	C. H. Mauney, Attn: J. D. Williams, 1739
1750	J. E. Stiegler, Attn: M. J. Eaton, 1759
1754	I. G. Waddoups, Attn: J. L. Todd, 1754
1758	C. E. Olson, Attn: D. D. Boozer, G. A. Kinemond, 1758
1760	J. Jacobs, Attn: M. N. Cravens, J. M. deMontmollin, 1760A
1761	T. A. Sellers, Attn: A. E. Winblad, J. L. Darby, 1761
1762	H. E. Hansen
1765	D. S. Miyoshi
4400	A. W. Snyder
4410	D. J. McCloskey
4413	N. R. Ortiz
4414	D. E. Bennett
4414	S. L. Daniel

DISTRIBUTION (Cont)

4414 M. S. Hill
4414 G. B. Varnado
4416 L. D. Chapman (5)
4416 K. G. Adams
4416 J. A. Allensworth
4416 H. A. Bennett
4416 D. Engi (10)
4416 L. M. Grady
4416 C. P. Harlan
4416 R. D. Jones
4416 M. T. Olascoaga
4416 C. J. Pavlakos
4416 D. W. Sasser
4416 D. R. Strip
5000 J. K. Galt
5600 D. B. Shuster, Attn: A. A. Lieber, M. M. Newsom, 5620,
R. C. Maydew 5630
5640 G. J. Simmons, Attn: R. J. Thompson, 5641,
L. F. Shampine, 5642
5641 C. A. Morgan
5642 B. L. Hulme
8266 E. A. Aas
3141 T. L. Werner (5)
3151 W. L. Garner (3)
For: DOE/TIC (Unlimited Release)
3154-3 R. P. Campbell (25)
For NRC Distribution to NTIS